NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IMPROVING FACE VERIFICATION IN PHOTO ALBUMS BY COMBINING FACIAL RECOGNITION AND METADATA WITH CROSS-MATCHING

by

Khoubeib Bouthour

December 2017

Thesis Advisor: Marcus Stefanou
Co-Advisor: Lyn Whitaker

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 2017	3. REPORT TYPE AND DATES COVERED Master's thesis

4. TITLE AND SUBTITLE IMPROVING FACE VERIFICATION IN PHOTO ALBUMS BY COMBINING FACIAL RECOGNITION AND METADATA WITH CROSS-MATCHING	5. FUNDING NUMBERS
6. AUTHOR(S) Khoubeib Bouthour	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number __NPS.2017.0066__.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Facial recognition is an important tool used by many disciplines, but its wider use in face detection and identification tasks has been somewhat limited. This is due to the many uncontrolled factors affecting faces in images, such as lighting, orientation, hair obscuration, blur, and the effects of aging. Despite tremendous efforts to overcome these uncontrolled factors, the reliability of a computer-based face recognizer is still questionable. In our research, we address the possibility of improving face verification using weighted cross-matching, which relies on a face verification metric and metadata. The idea is to implement a framework compatible with multiple platforms and capable of operating with limited resources while achieving satisfactory performance. We do not use statistical models, and we do not create patterns that require supervised learning. Our methodology is intended for use in personal digital image libraries because these libraries represent naturally context-correlated datasets. We use the native connection between files to determine the trustworthiness of an image relative to another. We then use this metric to attribute weights to pre-identified faces that are used as cues to help verify ambiguous elements. The final algorithm does not require the user's collaboration and performs automated digital image library management.

14. SUBJECT TERMS metadata, facial recognition, face verification, OpenFace, cross-matching	15. NUMBER OF PAGES 87
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

IMPROVING FACE VERIFICATION IN PHOTO ALBUMS BY COMBINING FACIAL RECOGNITION AND METADATA WITH CROSS-MATCHING

Khoubeib Bouthour
Major, Tunisian Air Force
B.Eng., Tunisian Air Force Academy, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 2017

Approved by: Marcus Stefanou
Thesis Advisor

Lyn Whitaker
Co-Advisor

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Facial recognition is an important tool used by many disciplines, but its wider use in face detection and identification tasks has been somewhat limited. This is due to the many uncontrolled factors affecting faces in images, such as lighting, orientation, hair obscuration, blur, and the effects of aging. Despite tremendous efforts to overcome these uncontrolled factors, the reliability of a computer-based face recognizer is still questionable. In our research, we address the possibility of improving face verification using weighted cross-matching, which relies on a face verification metric and metadata. The idea is to implement a framework compatible with multiple platforms and capable of operating with limited resources while achieving satisfactory performance. We do not use statistical models, and we do not create patterns that require supervised learning. Our methodology is intended for use in personal digital image libraries because these libraries represent naturally context-correlated datasets. We use the native connection between files to determine the trustworthiness of an image relative to another. We then use this metric to attribute weights to pre-identified faces that are used as cues to help verify ambiguous elements. The final algorithm does not require the user's collaboration and performs automated digital image library management.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AUC	area under the curve
BIA	balanced inter-class accuracy
BSR	best success rate
CFS	confirmed faces subset
CIF	currently inspected face
CMO	cross-matching output
CZ	confusion zone
DFS	discarded faces subset
EPR	equal positive rate point
ESR	equal success/error rate
EXIF	exchangeable image file format
FN	false negative
FP	false positive
FPR	false positive rate
FV	face verification
FVO	face verification output
HIP	histogram intersection point
LFW	labeled faces in the wild
NTP	nearest threshold to perfect classification point
OCT	OpenFace classification threshold
PDIL	personal digital images library
RI	reference image
ROC	receiver operating characteristic
TCI	trusted classified images
TN	true negative
TNR	true negative rate
TP	true positive
TPR	true positive rate
UFF	the user’s friend or family member
WCM	weighted cross-matching

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I want to express my gratitude to Dr. Marcus Stefanou, who believed in this project and dedicated time and effort in addition to providing advice and guidance in a friendly and professional atmosphere.

I am also grateful to Dr. Lyn Whitaker. In a few words, it was a privilege to have her as a teacher and advisor. Her insight inspired me and drove this research to a concrete finish.

None of this would have been possible without the support provided by Dr. Jeffrey Haferman and Mr. Bruce Chiarelli. Their inestimable expertise on Hadoop distributed storage cluster and MapReduce saved me weeks of testing and processing.

Finally, I want to express my gratefulness to my family and, particularly, to my wife, Soumaya, who sacrificed her education and profession to provide me with the best conditions to deliver the best of myself and compensate for the tasks that I failed to fulfill with my kids.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Since the rise of digital imaging, personal digital images have been extremely valuable to their owners. People enjoy taking, storing, and displaying photos to mark events and invoke memories. Now that such operations have become inexpensive, they have led to a huge increase in the number of digital images. It is estimated that 2.32 billion people in the world own at least one smartphone [1]. The combination of people's desire to record memorable moments of their lives and the impact of social networks and apps has caused the size of the libraries to become so large that it requires automated storage management. Manual management is time consuming, exhausting, and can lead to duplication and non-effective organization. Given the rise of pervasive cloud solutions, greater storage will undoubtedly be allocated to these digital image collections. In other words, digital album management is not only a personal issue. Leveraging its impact will contribute to a better user experience and optimized resource management.

The most common applications of personal digital images library (PDIL) management rely on the time and location of the image at the time of capture to group the files and display the hierarchy in a timeline format, a map spreading shape, or a combined representation. Location tagging of images has become available through GPS and was previously available using the General Packet Radio Service (GPRS) capability in modern cameras and smartphones [2]. However, what relates a user to a memorable event is not only the time and the place. It is a combination of *when*, *where*, *what*, and *who*. A study conducted by Wagenaar [3] shows that when recalling events, people tend to remember the people participating in the events. However, incorporating more means of organization will provide users with a better experience managing their PDIL. In addition, it is more convenient to group the images by people or relationships (for example, family, coworkers, friends, etc.) for future sharing and, eventually, privacy concerns.

For a system to be able to provide a people identification capability, the key tool is face identification, which is preferably called facial recognition. Consequentially, facial recognition has become a heavy field of research and has made great progress

throughout the last decade when many techniques were developed. The perpetual issue confronted when implementing a facial recognition system is the necessity to operate in a "constrained" environment. "Constrained" means that many conditions contribute to the success of the classification, such as face alignment, pose, expression, lighting, and aging effects. When such assumptions about these constraints are enforced, facial detection and recognition can beat the majority of techniques used in biometrics [4], particularly because they do not require the subject's cooperation. Unfortunately, most of the time images in a PDIL are naturally unconstrained. People might appear to be smiling, tilting their faces, or using accessories like glasses. Thus, facial recognition by itself is not efficient and does not guarantee the desired PDIL's management, which might lead to unacceptable misclassification rates. Therefore, the use of other cues is mandatory in order to improve the accuracy of the classifier.

A. PROBLEM STATEMENT

Most of the approaches that improve face recognition rely on machine-learning algorithms and require supervised learning, which implies image annotation and the user's interaction to establish the ground truth. In addition, these algorithms build models to perform predictions, which can be computationally expensive, and cannot guarantee extremely accurate classification. In this thesis, we investigate whether simple techniques involving a face recognizer algorithm and existing metadata can help reach acceptable classification performance without deploying statistical tools and without creating models and patterns. There is a tradeoff between the accuracy of the classification and the user's interaction. In this research, we evaluate the accuracy of the face recognizer without preexisting annotation. Such a method will fit in portable devices and personal computers, and can be adapted to cloud storage.

Imagine the scenario where a person owns a PDIL. The PDIL can be stored on a personal computer, the cloud, or, more likely, on the user's smartphone. Naturally, a person would tend to share his or her albums with family and friends. When sharing such files, the focus is more on either the events or the people. In our scenario, we consider the case in which a friend goes to the PDIL's owner and asks the owner to share the images

where he or she is present. We assume that the images in our corpus are not annotated (faces are not tagged) and not necessarily constrained. We want to imitate the human's behavior in such a task.

A human would scan the entire corpus, image after image, visually searching for that person. The human is also capable of discarding great portions of images if he or she knows that the desired person is not involved in that event, place, or at that period of time. He or she is also smart enough to identify the person even if that person's face is obscured, tilted, or blurred using high-level cues like other related people, clothes, hair, or pose. We state the problem as follows:

- Is it possible to improve facial recognition accuracy using simplistic methods and tools without needing the user's interaction?
- Can we develop a system to identify images of a person using only a new image of the face?

B. RESEARCH QUESTIONS

Our objective in this research is to investigate the possibility of improving face verification using free and available tools: the face verification algorithm itself and the image metadata. We want to find ways to achieve a higher success rate by adjusting the facial classifier. In addition, we want to confirm whether image metadata like time, location, and camera type can be used to infer the identity of the people on the image.

The goal is to run this method on an unconstrained personal photo album as that is the natural and most common status of images. The implementation is in an unsupervised environment in the context of automated annotation. We combine multiple improvement methods at both the pixel and content level along with the context level. The idea is to implement a weighted cross-matching framework where the main classifier is still the face verifier. We perform pairwise comparisons between each candidate (unclassified face) with all previously well-classified faces. The comparisons apply an attraction to the default face verifier to "push" the decision to either directions of the threshold as in [5]. This attraction is weighted based on the similarity observed in the metadata, which represent the context.

The following research questions guide this research:

- In the absence of user-annotated images, how can we use the same face verification algorithm to refine the results without any additional tools?
- What techniques can be used to improve the facial recognition accuracy, keeping the process automated and without any user interaction?
- Is there a way to combine context and content-based improvements for a better success rate using computationally light algorithms?
- Can the metadata be used to help the classification without creating profile patterns?
- How can the raw metadata contribute to an improvement without clustering or referring to any ground truth provided by the user?
- Can the metadata lead to a higher success rate by simple pairwise comparisons instead of per-user pattern?

Finally, we test and measure the performance of such techniques and the portability of the techniques across platforms. The method is implemented on a personal computer, but the data model and the training is performed on a Hadoop distributed storage cluster named GRACE on the NPS campus.

C. THESIS ORGANIZATION

In the next chapter, we discuss the state-of-the-art in this field and the different approaches already attempted. We highlight the differences and the similarities with our method.

The third chapter presents the description and implementation of our approach. We provide a definition of the dataset, the equipment, and software. In the same section, we clarify the reasons for all the decisions concerning the algorithms adopted and explain the boundaries of our work, the assumptions, and the open source/available resources/other research works involved.

In the fourth chapter, we deliver the results of our experiments. This section also includes the obstacles and eventual deviations made to overcome technical and

conceptual issues. We attempt to answer the research questions and show evidence for our interpretations. In this chapter, we also discuss the efficiency and the portability of the final algorithm.

Finally, in the last chapter, we summarize the overall experiment and show the benefit of such technique and its contribution to the current industry. We discuss the impact on the final user and on IT companies. We also discuss combination with other previously implemented methods and eventual extensions as future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. FACE VERIFICATION REVIEW

Identifying people in images or videos involves facial recognition techniques. Facial recognition is a general topic that includes face identification and face verification. The difference between these two similar techniques is that identification is the process of attributing a name to a person inferred from multiple sources of information that might or might not include a description of the face features. Verification, also called authentication, is simply the ability to determine whether two faces represent the same person or not. The usability can be different, but both techniques rely on computer vision techniques that extract the face's features, transform them into a logical and numerical representation, and measure the similarity to state the decision output.

1. Face Representation

Coding faces into a data structure involves transforming the shape, color, texture, and brightness from the photo representation into a digital and exportable format for future manipulations, such as measures of similarity. This transformation is not a recent concern. In 1991, Turk and Pentland [6] studied the aspects of the faces for detection and coding, creating a framework for facial recognition with principal component analysis using eigenfaces and L_2 distance between pairs of images. Figure 1 shows the parts of the face considered during the transformation process.

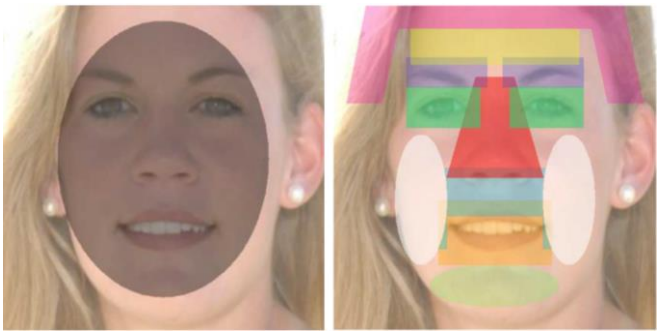


Figure 1. Face Regions for Automated Feature Selection on Aligned Face Image. Adapted from [7].

Other techniques attempt to transform a digital face image into a particular data structure allowing manipulation and comparison, such as enhanced Fisher linear discriminant analysis approaches [8], local quantized patterns [9], and locally adaptive regression kernels (LARK) [10]. A large part of these techniques first spot landmarks or features and then turn them into some metrics to be used for description.

Other approaches run template matching and use deep neural networks relying on large face datasets. A team from Google developed a system called FaceNet [11] that uses a deep convolutional network to transform a face to a features hyperspace of 128 bytes. By applying this representation, Schroff [11] achieved high success rates on most known benchmarks. Amos [12] used the work of Schroff [11] and developed OpenFace, which translates a face into a 128-dimensional unit hypersphere. Measuring the distance between two faces provides an output from zero to four representing the similarity and allowing not only classification but also clustering.

2. Challenges

Facial recognition operates better in a constrained environment, where faces obey certain assumptions about conditions including lighting, brightness, pose, expression, and hair obscuration. In real life, images are not always taken under controlled conditions, and the faces are generally unconstrained. Figure 2 illustrates the case where a face is partially covered by the subject's hair.

Figure 2. Partially Covered Face with Expression. Source: [13].

Facial recognition loses its efficiency when these assumptions are not met. Nevertheless, it is the main tool to infer identification from the content. Extra refinement must take place to reduce the impact of lighting and makeup (image processing), pose (face alignment), and expression.

In previous research, scientists improved face verification from different perspectives. From the pixel level to the content to the context or a combination of different methods, the objective was always to create a workable framework capable of handling unconstrained images and to get as close as possible to the human performance using reasonable resources and being power efficient. Content-based improvements are techniques applied to the content of the images to infer the identity of the people, like hair recognition. Context-based improvements are methods that involve only information that is not related to the content like spatial or temporal re-occurrence. Most of these techniques help overcome some of the obstacles and might contribute to obtaining a quasi-constrained data. However, not all help with problems related to partial obscuration or non-frontal faces. Table 1 lists the best-known techniques proven to increase the success rate of facial recognition.

Table 1. Categorization of Different Improvement Approaches

Content-level	Context-level
- Pixel level Color/Brightness correction 2D / 3D alignment - Hair recognition - Expression pattern - Attribute matching - Pose pattern - Clothes invariant	- Co-occurrence (clustering subjects) - Popularity (frequency of reappearance) - Re-occurrence (probability of appearing at the same period/location)

For more information on the different approaches, please see [5], [7], [14], [21].

It has been proven that human performance drops in face verification when restricting the process to the face and eliminating the surroundings. Figure 3 shows the impact of the surroundings on the human being's performance. The red curve represents

the performance of a human when looking at the whole image. That performance drops when the faces are cropped (blue curve). Knowing that computers see faces in the cropped format explains the necessity of providing the algorithm with additional cues. The whole environment including context is necessary to achieve high success rate, and this is done by humans intuitively. As the science of computers is the science of modeling real-world behavior in information systems, researchers look for more intelligent and augmented techniques capable of providing classifiers with high-level cues.

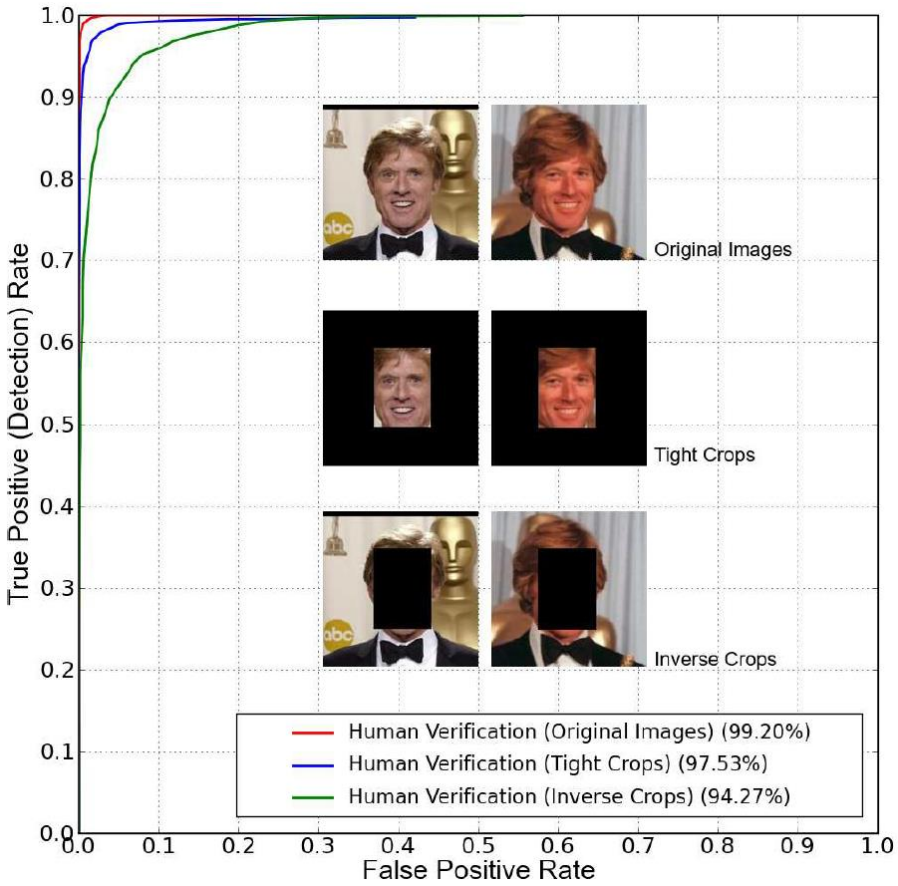


Figure 3. Face Verification Performance on LFW by Humans and Surrounding Impact. Adapted from [7].

B. IMPROVING FACE VERIFICATION

The face verification improvement efforts made by researchers can be grouped in two categories:

- Content-based improvement: relies on visual cues found in the image
- Context-based improvement: uses the metadata, the user's annotations, or any existing indicators (for example profile patterns or probabilistic models) to assist the classifier determine the appropriate decision.

1. Content-Based Improvement

Given the previously cited obstacles, other approaches were investigated over the years to overcome the deficiency of facial recognition. These approaches deal with situations when the subject is not following the desired and suitable pose conditions, the image resolution is poor, or even when the classifier is confused and might require additional information to adjust the decision. Among these approaches, we examined content-based improvement based on cues given to the classifier derived from the image and its content. The user annotation might be used in this context to represent the ground truth for training, as most of these techniques perform supervised machine-learning. Below, we enumerate some approaches that fall under the content-based category.

a. Aligning Faces

2D and 3D alignment can be applied to faces in order to reduce the impact of pose during the phase of preprocessing the data. It is proven that aligning faces has a great impact on the efficiency of classification as it reduces the dissimilarity from a computer vision perspective [14], [15]. Nevertheless, it does not entirely solve the issue, as many other factors, such as lighting, are not affected. In fact, running such scripts on the Labeled Faces in the Wild (LFW) benchmark dataset [22] failed to demonstrate significant improvement using the state-of-the-art facial recognition. Various 2D alignment methods have been applied to LFW [7], but none represents a global solution as they are tightly coupled to the dataset.

Aligning faces is computationally expensive [7]. That cost increases when attempting to improve the accuracy of the alignment by avoiding its application at the preprocessing phase and executing it at each pairwise comparison, trying to align faces to each other, instead of using a common coordinate system [16].

Processes like face alignment and color correction have become part of all recent face detectors and recognizers. As they contribute to obtaining higher quality faces but are not part of the classification, these methods are considered low-level in a logical context (does not imply low impact). Therefore, these methods are embedded in the process of adapting faces as part of the preprocessing phase to allow better classification based on more reliable data. Figure 4 shows how face detection and alignment are executed at the early stage before the classification takes place.

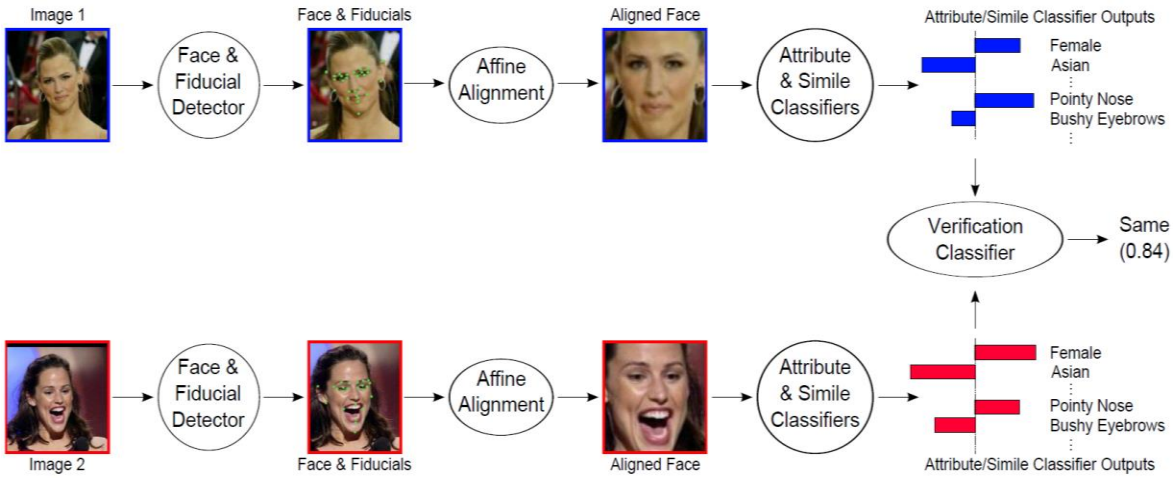


Figure 4. The Face Verification Process. Adapted from [7].

b. Hair Recognition

Hair is also used by many researchers for people identification. Roth and Liu [17] show the usefulness of their framework against the LFW dataset. First, the hair is part of the head. Thus, hair recognition can be executed as long as a face verification is applicable. In contrast, body recognition cannot always be performed unless the whole person is captured. Second, a human usually has a consistent hair appearance that changes infrequently. Nonetheless, the subject can change the color of the hair or the cut, which would affect the classifier. This cannot be considered a major obstacle as, at most, the program groups the subject's identified faces into two or more groups based on the appearance. This is still manageable if the intent is to provide the user with an optimized short list of suggestions for annotation.

The hair can also be a good visual distinguisher between humans. The color, length, and style are good discriminant factors for a good classification. Figure 5 shows a high-level description of the process of hair verification, which is similar to face verification. Unfortunately, not all people have hair, and this feature also presents many similarities. In addition, it is challenging to distinguish the hair from the beard in many cases, thus adding an extra layer of complexity. Nevertheless, hair recognition can be used as a cue and integrated as part of the facial recognizer to provide additional information for a better classification.

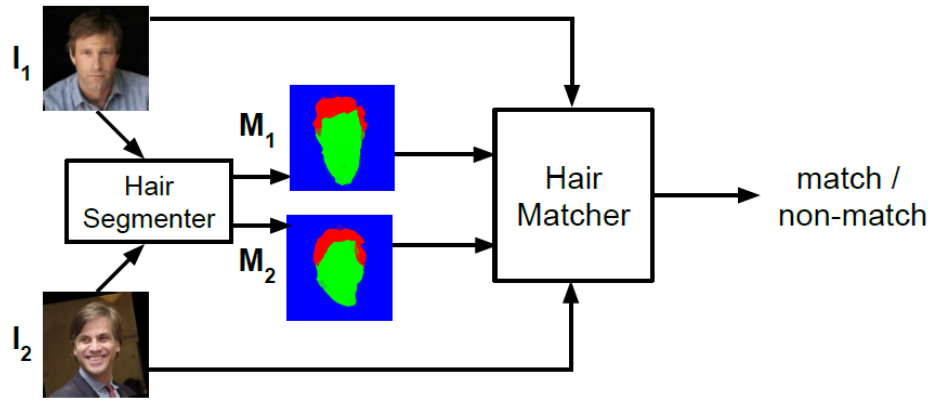


Figure 5. Example of Hair Extraction and Analysis. Adapted from [17].

c. Expression Recognition and Attributes Matching

The main objective of such approaches is to address the limitation of face verification against facial expressions and to use multiple attribute classifiers to narrow down the list of candidates for the classification. Kumar [18] uses a framework for detecting the gender, race, age, skin, and hair color along with other visual features but isolated from the global face description. This helps overcome problems in an unconstrained environment, as most of these attributes will match. In Figure 6, we see a set of features used by Kumar [18] in his framework for attributes matching. Note that 14 out of 16 attributes match, although the lighting and pose are different. Some of these attributes are already part of the facial representation, but exporting them as separate

attributes results in assigning more importance to their output. This process is proven to increase the face verification success rate.

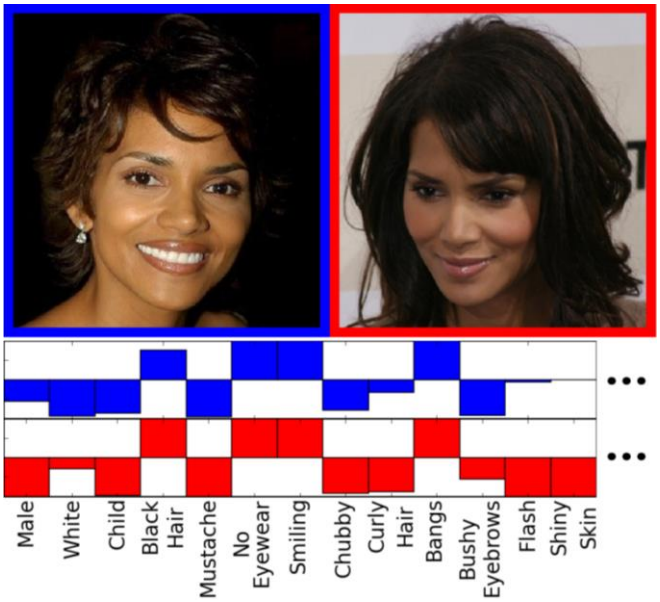


Figure 6. Attribute Classifier. Adapted from [18].

Similarly, Kumar [18] implements an expression verifier by combining multiple variables to represent a facial expression and using them separately to measure the similarity of smiles on two faces. Both approaches result in an increase in the success rate on the LFW dataset compared with the current state-of-the-art recorded performance. These methods are noticeably similar to the previously discussed improvement approach, hair recognition, since they can always take place whenever facial recognition is performed given that such techniques are based on the co-locality with face region.

d. Clothes Invariant

Clothes are also a good cue for face verification as they are considered part of the person's identity. Li [19] discusses how clothing and body appearance can compensate for the facial recognition limitations, especially when the face is obscured. His objective is to detect personal patterns and "exploit higher intra-personal appearance consistency within photo groups" [19]. The limitations of such methods are when the image is limited

to the face or the subject is not covering his or her upper body. Furthermore, in environments like sports or the military, this can create similarities and therefore penalize the algorithm. Lastly, a person can appear with a new look in different albums.

e. Pose Pattern

This method is restricted to the cases where the subject's body is captured in the image. It demonstrates great improvement in the field of Face Verification (FV), beating the most sophisticated modern techniques including DeepFace [20] when used on its own since this last technique cannot operate well on images in the absence of faces. Zhang [5] uses the concept of Pose Invariant PErson Recognition (PIPER) using poselets [23], where the face is considered a particular case. Poselets are classifiers describing a pose pattern. Zhang [5], by attributing patterns to identified people, is able to verify faces by augmenting the recognition capability via the pose classification. As people tend to have their own customized pose pattern, the researchers use it as a cue to identifying the person. Supervised machine-learning algorithms are used to detect and create these patterns and profiles.

f. 3D Extraction

3D alignment is introduced to overcome the variance in pose and provide better representation than the 2D process. In Figure 7, image (b) represents the 2D alignment of the face on (a). On (g), we observe the improvement made possible by applying 3D alignment. 3D extraction uses stereo images to create a 3D model that can later be transformed to fit desired position, limiting the difference in alignment. Blanz [15] uses 3D feature extraction as a means of classification using the shape and the texture.

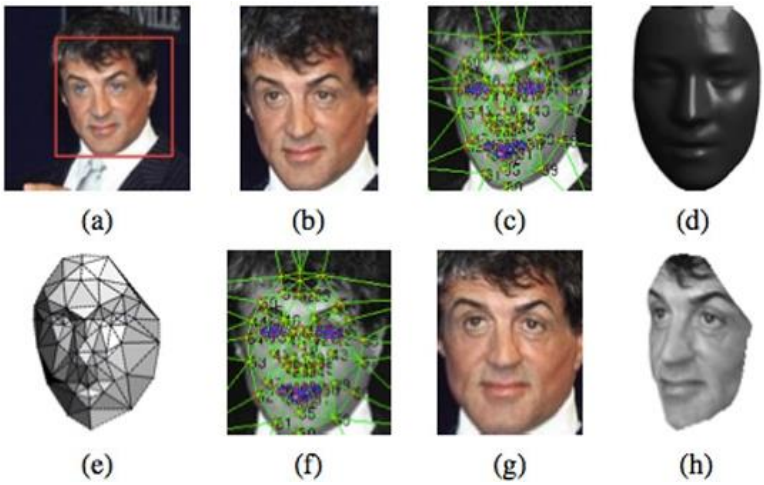


Figure 7. 3D Alignment Pipeline. Adapted from [20].

In 2014, Facebook announced a new system called DeepFace [20] and proclaimed that they approach the human being's performance in face verification. In his paper, Taigman [20] claims that their system is not only very accurate but, more importantly, it has a 97.25% success rate on unconstrained images, while the recorded performance of human beings is 97.53%. Moreover, Taigman [20] confirms that the system is highly scalable and can handle a tremendous amount of images.

Stating that face recognition consists of four main stages, detect ⟹ align ⟹ represent ⟹ classify, the Facebook team's intent is to revisit the alignment and representation steps. They use a 3D face modeling for alignment and a nine-layer deep neural network for representation. Taigman [20] also admits the use of a huge corpus of images for training, which cannot be made available to other companies or platforms. Facebook also relies on the intuitive cooperation of the users who tag themselves and their friends and provide the researchers with gorgeous datasets for training.

g. K-Nearest-Neighbors

With the existence of distance-like measure of similarity between two faces, it becomes easy to implement techniques for grouping and clustering. Guillaumin [24] adapts this tool with a K-nearest-neighbor algorithm and shows an increase in the success rate of face verification applied to the LFW. Guillaumin [24] uses existing labeled faces

and calculates probabilities based on the similarity distance. Comparing his technique to the actual state-of-the-art, Guillaumin [24] demonstrates an improvement on both the constrained and unconstrained settings.

Similar work made by Zhang [25] aims to perform automated annotation based on labeled images and then applies a K-nearest-neighbor technique to improve the classification of the unannotated faces. We note that such techniques rely heavily on user-annotated faces and use machine-learning algorithms for training and parameters determination (thresholds and minimum number of votes).

2. Context-Based Improvement

This form of seeking to improve face verification accuracy addresses the metadata of images. Metadata is data about the data. In our case, it covers information such as time of capture, camera, resolution, and in some cases the location (GPS), altitude, and orientation. Tags and labeled faces can be included in the metadata. Metadata extends to higher level information like social activities and events. Nevertheless, these precious cues are not always accurate and available. Modern cameras and smartphones have the GPS capability and can timestamp the images. However, conclusions about social activities require high-level artificial intelligence applications and/or the user's collaboration.

a. Temporal and Spatial Re-occurrence

Temporal and spatial re-occurrence are the first fields explored by researchers. They are the most autonomous and reliable information contained in the metadata. The idea behind this approach is that a person appearing in one image is very likely to appear in another image taken at the same location or within a short timeframe from another image where that person was previously identified. For example, family members tend to appear more in images taken at home, whereas coworkers are more likely to appear at the office. In another example, colleagues are likely to appear during office hours. Naaman [21] starts from user-annotated images to build patterns for identified people. Naaman's [21] objective is to reduce the heavy load of annotating images on the user by reducing the list of suggested people to a reasonable length based on probability.

b. Co-occurrence and Social Connection

Co-occurrence addresses the likelihood for a person to appear in an image, knowing that another person was previously identified [21]. In other words, if two or more people are identified in an image, it is very likely that they will show up together on another image taken at the same location or time. This phenomenon can be extended to other events and help attributing probability based on the personal patterns. This concept leads to the notion of events and social context. By investigating the connection between people, Naaman [21] succeeds in attributing ranks to expected output based on social relationship with the annotated faces.

c. Popularity

This topic is developed in [21], where the researchers state that some people are more frequently identified than others. Such a conclusion can help attributing weights wisely to the classifier when it is confused by the similarity between two candidates.

C. PEOPLE IDENTIFICATION IN REAL LIFE

Multiple facial recognition applications for PDIL management exist. For example, Apple embeds a facial recognition application in its "Photo" app as part of the iOS. It allows the user to add annotation to the faces, as shown in Figure 8, and can infer the identity from the contact's picture.

Available at https://support.apple.com/en-us/HT207103

Figure 8. Apple iOS Photos App: People Identification and Grouping.

Google also implements its own people identification application integrated with Google Photo. Both Apple and Google offer more than people grouping. They provide location and date grouping as well as events and activities. These companies are able to detect the user's location not only from the GPS coordinate but also inferring it by looking at the content of the image. They can discern activities such as whether a person is on the beach or boarding an airplane.

Facebook dominates this field. Possessing the largest user-annotated images dataset, they have proven to be able to achieve a very high success rate in identification in unconstrained conditions. Yann LeCun, an expert in computer vision and pattern recognition who works for Facebook, said in a conference in Boston in 2015: "There are a lot of cues we use. People have characteristic aspects, even if you look at them from the back." He also stated that their algorithm can accurately identify a person in one photo out of 800 million images in less than five seconds [26].

In 2016, Amazon announced Amazon Rekognition. This is an API that allows users to integrate image analysis with their applications. Amazon states on their website: "With Rekognition, you can detect objects, scenes, faces; recognize celebrities; and identify inappropriate content in images.". In the same fashion of previous approaches, Amazon uses deep neural networks for classifications. Their service is good for near real-time batch treatment, grouping, and classification. Unfortunately, Amazon does not publish any performance results of their product on known benchmarks in the field such as LFW.

D. THESIS MOTIVATION

Considerable efforts have been made in the field of face verification and great progress has been witnessed. Machine performance has come close to human performance. Nevertheless, facial recognition lacks accuracy when operated in an unconstrained environment. Our research is a continuity of the work of Naaman [21] combined with the method used by Zhang [25] with a few differences. The success of Naaman [21] and Zhang [25] derive from previously annotated faces, which allow patterns and profiles to be built. Their goal is to deliver a short list of suggested people

that reflects an improvement in the classification system. The method used by Naaman [21] is based on popularity, co-occurrence, temporal and spatial re-occurrence, whereas Zhang [25] mainly uses a facial recognition algorithm in addition to the k-nearest-neighbor method. Our method consists of using the same contextual information–the metadata–and combines it with a facial recognition algorithm to perform automated annotation on unconstrained and unannotated PDIL.

We build a framework that addresses the success rate of face verification but constrained to the classifier and the metadata only. There is no ground truth and no user-annotated images. The system is mainly unsupervised. We execute three similar experiments and generate results for comparison. The first involves only a facial recognizer. In the second, we introduce cross-matching where we perform a pairwise comparison between faces and adjust the classifier's decision based on the similarity. Finally, we extract the metadata correlation to attribute weights to the previous similarities for a higher-quality influence on the decision.

Our approach's goal is not to improve how face detection, face description, and face representation operate. We do not discuss tuning existing algorithms. Any performance and accuracy improvements regarding the detection, alignment, description, and distance measure of faces is not in the scope of this thesis. We use off-the-shelf open-source libraries, and we focus on improving the success rate by implementing our technique.

For privacy reasons, only the author's images are used in this research. For other profiles, fake names are attributed and their images are not displayed. All that is needed is the digital representation of the faces, which is a function performed by OpenFace [12].

III. METHODOLOGY

A. USE CASE

We consider the scenario when the user owns a PDIL and someone asks the owner to share the images in which he or she is present. We call that person the User's Friend or Family member (UFF). We do not assume that the images are annotated at the time of capture. We are more concerned about scanning the entire photo album using computer vision techniques to locate images as a human would do in the real world.

In order to operate, the system needs to know what it is supposed to find. We need to feed the program with the UFF's image to be used for verification. This source can be retrieved from the library, from the contacts application, or simply by a new capture. This input is addressed as the Reference Image (RI).

Two considerations are made regarding the RI. First, since the user is interacting with the system at this stage, we assume the RI obeys a good standard for face verification. In other words, the lighting, pose, expression, and the majority of the constraints are controlled to avoid over-penalizing the classifier which suffers from the unconstrained characteristic of the dataset. Second, we assume that there is nothing connecting the RI to any of the existing images in the photo album. If the RI derives originally from the PDIL, its context—the metadata—is ignored. Figure 9 shows an example of RI fed to the analyzer and satisfying the requirements for facial recognition.

Figure 9. High-Level Illustration of the Face Verification Process and Output.

B. IMPLEMENTATION SUMMARY

In this method, we do not only use face verification. In fact, the research community agrees that face verification does not work well on unconstrained images, which is the case in this study. Many improvement methods exist, but our goal is to investigate the impact of what we call Weighted Cross-Matching (WCM). We define cross-matching as the concept of comparing each face to all faces that were previously classified with a high level of confidence, called Trusted Classified Images (TCI). Next, we use the output of face verification between the TCI and the current face to apply an attraction to the decision and push it to either side of the threshold. For instance, a face that we fail to determine as being our UFF might be highly similar to a group of TCIs, thus forcing reconsideration of the initial decision. Similar approaches have been taken under this same idea. Zhang [25] relies on groups of user-annotated images, which are used as ground truth. Moreover, the fact that he relies only on labeled faces implies that he compares the faces only against the true positives. In our approach, a TCI can be on either side of the threshold. It can be a highly similar face to our UFF, or a face that was discarded with high confidence. The notion of true positives and true negatives cannot be used in this context as the TCIs theoretically are not annotated, except for results measurement purposes in our experiments.

The WCM is developed in two stages. In the initial implementation, we run cross-matching without any weight adjustment. Later, we measure the appropriate thresholds to be applied to the distance metrics, creating the list of weights to be used during the cross-matching process. In this step, we consider the metadata of the images and look at the pairwise similarity (for example, were these two images taken with the same camera?). The only one-time offline training part in this process is to determine the thresholds needed for determining the similarity between images when comparing their metadata. We do not plan to create patterns and profiles, but we want to evaluate the distance when we refer to proximity in GPS locations in regards to the entire dataset and not to a particular subject or a pair of images. Based on the similarity, when performing WCM, an image can have higher impact than another on the currently analyzed face depending on the number of metadata features they share. Images taken within a short timeframe

and at very close locations are more likely to have the same person than images taken at the same location but not on the same date. This concept is discussed by Naaman [21], but his conclusion affects the probability for a particular person to be present on the image, so he adjusts the list of eventual candidates for annotation. This mechanism requires recording and following people's patterns. In our approach, the conclusion only affects the importance of the current TCI. This tells the classifier how trustworthy this reference is without the need of grouping or clustering, but simply measures the similarity and attributes a degree of confidence to references.

The pipeline of our approach can be described in two passes as follows:

detection → alignment → description → verification → cross-matching → classification

1st pass 2nd pass

Figure 10 shows how the output of the verification phase attributes a value to each face. Consequently, a face can be classified using predetermined thresholds. High-confidence level classified faces fall in the TCI zone. Elements in the confusion zone will be subject of further manipulations for better classification.

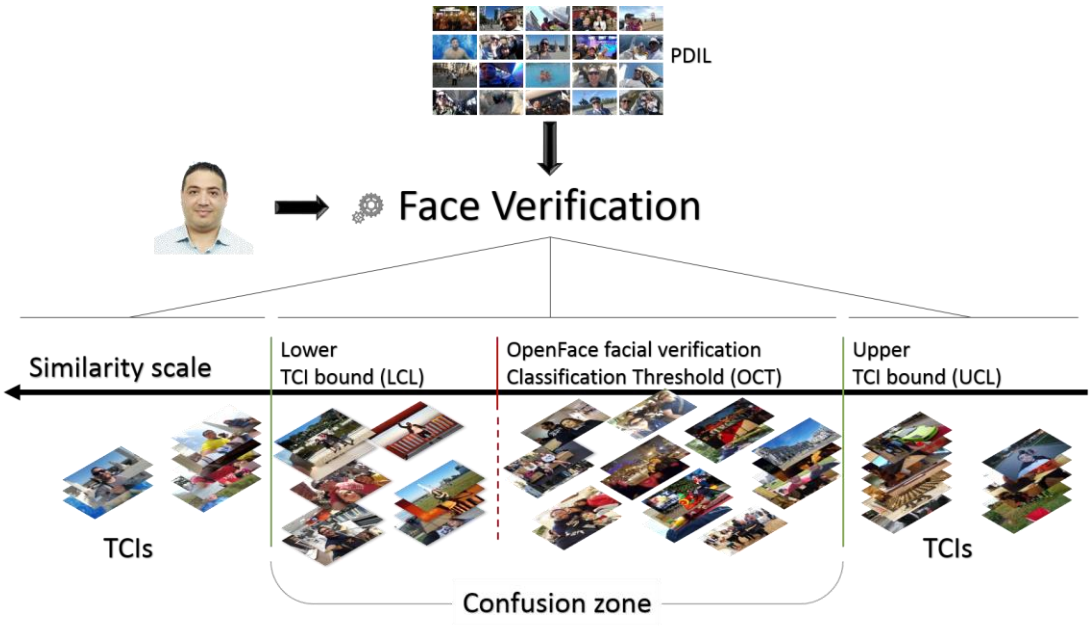


Figure 10. Face Verification Process: First Pass.

C. DATASET DESCRIPTION

To our knowledge, there are no publically available datasets that we could have used for this work. The specification of the images needed for our purposes is that they must be naturally related by the context (events and activities like trips or birthdays) to infer the identities of people. This can be found in PDILs. Datasets like LFW, used as a benchmark in most of the recent research, cannot be used in our context since the images do not represent any trivial connection between each other. In addition, our method is for a user-oriented usage. We keep our approach simple and implement it on portable devices as it relies on images captured by the cameras and not collected from other sources like public search engines.

In this research, we use the author's PDIL. It consists of a corpus of 24,346 images with a size of 96.5GB captured from 2011 to 2017. The images are spread across 95 folders without nesting. These folders are important as they form some sort of grouping. The resolutions of the images vary as they were not all taken by the same camera. Many images are missing geographic information since they were captured with cameras that were either without GPS capability or with the GPS feature disabled by the user. Table 2 is a summary of the dataset.

Table 2. Dataset Summary.

Number of images	24,346
Number of folders	95
Corpus size	96.5 GB
Images time span	2011-2017
Total number of faces detected	48,784
Real faces	27,115
Labeled faces	15,313
Images with real faces recorded	13,460
Images missing GPS data	2,537
Faces missing GPS data	5,004

D. DATA PREPROCESSING AND TOOLS SELECTION

Our intent is not to improve the verification algorithm itself. Instead, we use its output and introduce some adjustment based on existing metadata. Therefore, some manual cleaning must take place in order to prepare the data for analysis and interpretation.

Our experiment is performed in two passes. The first phase involves algorithms such as face detection, alignment, description, and verification. We use existing off-the-shelf tools for each process. Many implementations of each part of the procedure exist. While performance varies across datasets, from accuracy and execution time perspective, we are not concerned about the selection of the best existing techniques. Moreover, our approach might need to be tested with different algorithms to be proven efficient or not. For the scope of this thesis, we limit our choice to open source libraries.

For face detection, we use the Haar cascades detector described in [27], which is available in the OpenCV library. It allows detecting faces under controlled conditions. This method is considered because of its ease and speed of use. It does not perform any sophisticated tasks other than a simple face detection using a simple pre-trained tree-based algorithm that does not require the training of or use of machine learning tool. After running the Haar cascades over our dataset, we fail to detect the majority of faces. This is due to the nature of faces in the images, as they are unconstrained. Since our objective is not to improve the Haar classifier, we adjust the parameters to allow more false faces in order to collect as many true faces as possible. This flexibility yields 48,784 faces, but only 27,115 are true faces. We discard the false positives and label the rest of the data accounting for only 15 people that are more or less heavily present in images. These 15 persons, which we call profiles, stand for future UFFs. Only 18,431 images out of the original 24,346 end up having faces detected in them. Some images might have more than one face in them. In some cases, there are no faces. The number of images with true faces is important as it affects the complexity of the algorithm that we propose, particularly when we start the cross-matching process. Table 3 shows the distribution of the faces across profiles.

Table 3. Labeled Faces Distribution across Profiles.

Profile	Abrha	Astraat	Bonji	Heeda	Hickam
Count	667	26	70	33	88
Percentage	2.46%	0.10%	0.26%	0.12%	0.32%
Profile	Hmouda	Holu	Khufu	Laghbesh	Mekah
Count	5	19	2,227	14	179
Percentage	0.02%	0.07%	8.21%	0.05%	0.66%
Profile	Mimyth	Sakis	Sierra	Sokhoi	Yakouza
Count	1,074	29	3,412	2,634	4,836
Percentage	3.96%	0.11%	12.58%	9.71%	17.84%

The framework we intend to build is suitable for use with any viable face verification tool, as long as it provides a similarity metric. In our experiment, once a face is detected, we opt for the open source OpenFace library [12] for face alignment, description, and verification. We could have used the built-in face detector of OpenFace, but we opted for the Haar cascades face detector since it is significantly faster and we are not concerned by the efficiency in terms of detection accuracy. Nevertheless, the performance of the two is not very different since OpenFace uses OpenCV. We use OpenFace to take advantage of its deep neural network to generate a low-dimensional representation of each face that will eventually be used to measure distances between faces. Figure 11 shows the workflow of an image taken from the LFW dataset when processed by OpenFace. The output is a real-valued vector stored as a Numpy array of 128 float values.

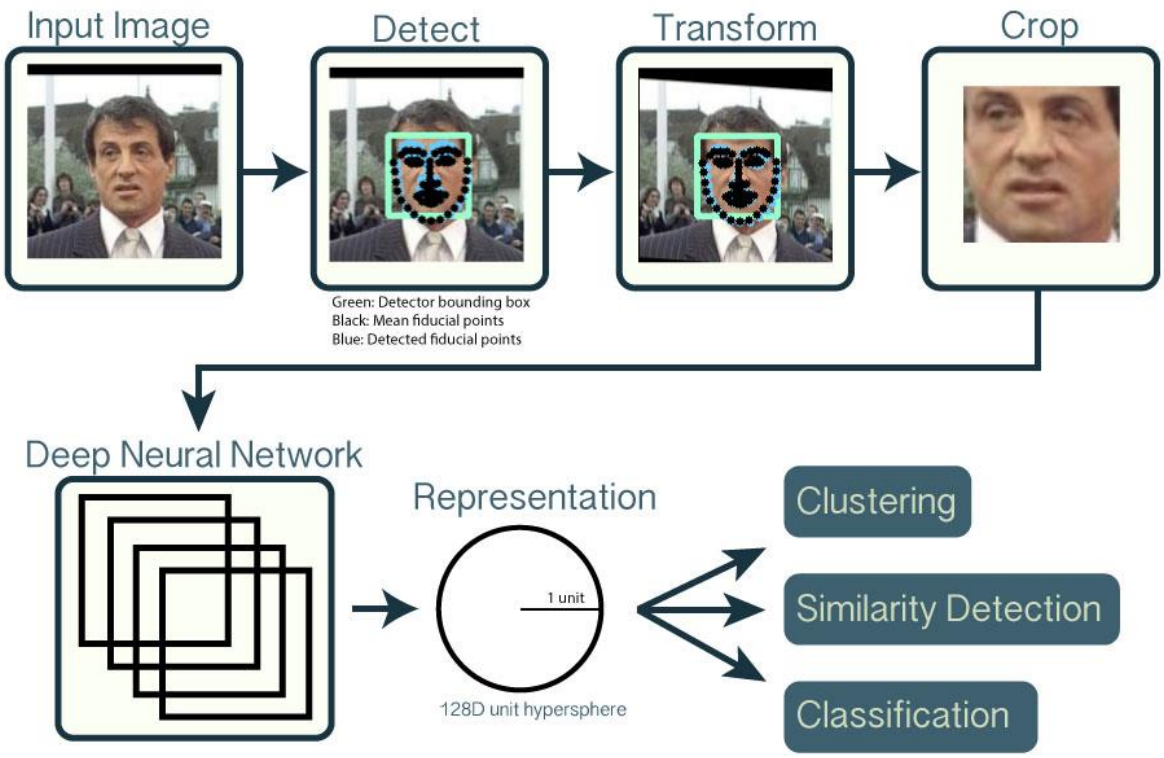


Figure 11. Image Representation by OpenFace. Source: [12].

E. AUTOMATED ANNOTATION

1. Experiment 1: Face Verification Only

We use the OpenFace library for face verification. This library offers several tools including face detection, alignment, transformation, and representation. We use just the latter part for face representation where every face is translated to a 128-float Numpy array. The distance between two faces is measured using the dot product of their matrices, which is scaled to take values between zero and four. The closer that value is to zero, the higher the similarity of the compared faces. The closer the value is to four, the more likely we have two different persons.

The OpenFace team recommends 1.0 to be the classification threshold. We conduct some measurements to find the suitable threshold for our dataset. This phase is not mandatory but is implemented to refine results and to investigate whether improvements can still take place although we use the best threshold. Default settings can be used without drastically affecting the output. Once the parameters are set, we identify

the 95% confidence level thresholds of true positives and true negatives. These are used for TCIs selection. We call them respectively Lower bound Confidence Level (LCL) for class one and Upper bound Confidence Level (UCL) for class zero.

This experiment is used as a reference for results interpretation since we cannot use the state-of-the-art benchmarks, or more specifically the LFW standard, because of the nature of our dataset, which is a PDIL with natural existing correlations.

2. Experiment 2: Cross-Matching

Much research, such as [25], implement techniques similar to cross-matching, but these techniques rely on user-annotated faces. These methods are based on the idea that the probability for a face to match the UFF is a combination of multiple classifications across a set of pre-annotated images of that same person. In our approach, the implementation is slightly different. The process is intended to be fully automated. Thus, no user interaction is involved, and we do not consider any pre-labeled image. In order to be able to apply cross-matching, we need to obtain sets of TCIs from the existing images. The output of OpenFace representation makes this possible. Since the exported value of that algorithm yields a distance between two faces, we can organize the PDIL and sort the faces based on that factor of similarity.

The first pass of our method, which consists of just the face verification, generates three subsets of images. The settings for that phase are discussed in the next chapter. The main parameters include the LCL, UCL, and the OpenFace Classification Threshold (OCT) used for the decision. Based on the OpenFace face verification output (FVO), the currently inspected image is attributed a category. If the FVO is less than the LCL, the face is considered a TCI with great similarity with the RI. This group is called Confirmed Faces Subset (CFS) since we are fairly confident that the face described is the UFF's face. Similarly, for an FVO greater than the UCL, that group is called the Discarded Faces Subset (DFS), where we are assured that in 95% of the cases that face is different from the UFF's face. We obtain a middle range of values. This last interval is called the Confusion Zone (CZ), where the system needs to perform deeper analysis to filter out the

false positives and false negatives. Figure 12 illustrates the different sections in the scale of the FVO. The values present on the figure are discussed in the next chapter.

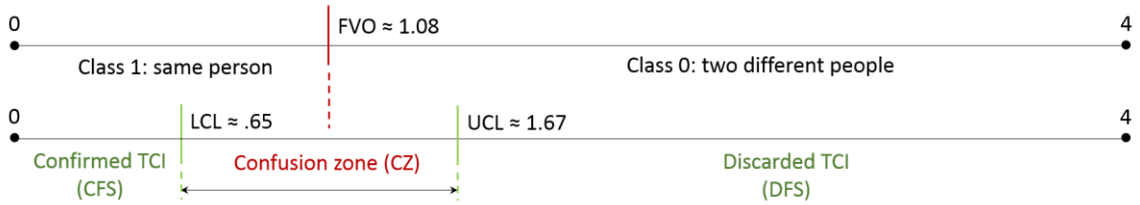


Figure 12. The Three Subsets Needed for Cross-Matching.

Once these three groups are created, the second phase can take place. The algorithm of cross-matching consists of taking each element of the CZ and running FV against all elements of the CFS and DFS, focusing only on the cases with high similarity or high dissimilarity. Only three cases are of interest:

- High similarity with an element of the CFS → attraction to class one: Type one force.
- High dissimilarity with an element of the CFS → repulsion to class zero: Type two force.
- High similarity with an element of the DFS → attraction to class zero: Type three force.

No other case is significant or can help with a decision. For instance, if a face from the CZ is highly different from a face from the DFS, we are unable to make any conclusion concerning the decision. In contrast, if the current face is different from an element of the CFS, it means that it is different from the RI, and the output is affected accordingly. A description is provided by Figures 13 and 14. Figure 13 shows how comparing a face to an element of CFS generates either attraction or repulsion. Whereas in Figure 14, we see how DFS contributes only to attraction.

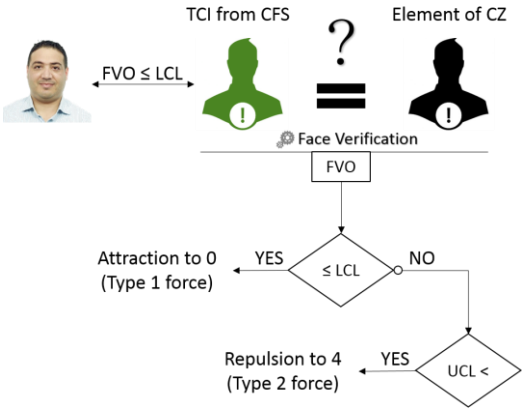


Figure 13. Comparing Element of CZ to Element of CFS.

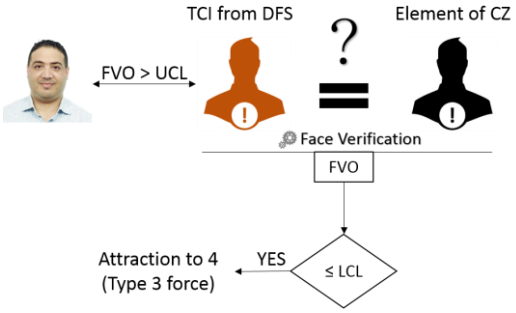


Figure 14. Comparing Element of CZ to Element of DFS.

Once the second pass completes, we obtain a list of force vectors to be applied to the initial FVO for each element of the CZ. Figure 15 shows an example of the resulting list and illustrates the force vectors that apply to the initial FVO.

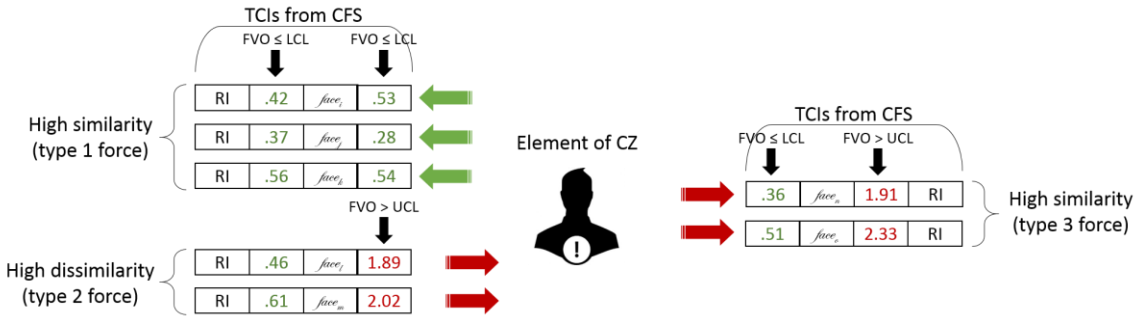


Figure 15. Illustration of the Three Types of Forces.

We use this list of force vectors to compute the center of mass and determine the new FVO. Generally, we expect to observe only unidirectional attraction. Nevertheless, we need to account for all scenarios. Therefore, when calculating the new center, the more observations of one type of force we find, the more weight we attribute. We also need to keep in consideration the importance of the initial FVO of the Currently Inspected Face (CIF) against the RI. For that reason, a relatively high weight is assigned to it.

Below is the mathematical formula applied during the cross-matching process and the calculation of the center of mass. We define ω the initial weight attributed to the FVO of the CIF compared against the RI. $\mathrm{FVO}(x, y)$ is the face verification output when comparing face x to face y. We define F_1 the set of all type one force vectors, future attractors to class one. Let m be the cardinality of F_1, i.e.:

$$m = |F_1|, \qquad F_1 = \{\text{all faces } e \in \mathrm{PDIL}, \mathrm{FVO}(e,\mathrm{RI}) \leq \mathrm{LCL} \text{ and } \mathrm{FVO}(e,\mathrm{CIF}) \leq \mathrm{LCL}\}.$$

Similarly, n is the cardinality of the union of F_2, the set of type two force vectors, and F_3, and the set of type three force vectors. These two sets hold the attractors to class zero.

$$n = |F_2| + |F_3|,$$

$$F_2 = \{\text{all faces } e \in \mathrm{PDIL}, \mathrm{FVO}(e,\mathrm{RI}) \leq \mathrm{LCL} \text{ and } \mathrm{FVO}(e,\mathrm{CIF}) > \mathrm{UCL}\},$$

$$F_3 = \{\text{all faces } e \in \mathrm{PDIL}, \mathrm{FVO}(e,\mathrm{RI}) > \mathrm{UCL} \text{ and } \mathrm{FVO}(e,\mathrm{CIF}) \leq \mathrm{LCL}\}.$$

To calculate the resulting FVO that we call Cross-Matching Output (CMO), a simple measure of the distance and sum of vectors across all sets is not adequate in the context of classification. The thresholds for both the CFS and the DFS are not symmetric from the OCT. Thus, the threshold that is at a larger distance from the current FVO of the CIF will have higher influence. Therefore, averaging the distances in each side is more appropriate to account for that difference in scale. We define α to be:

$$\alpha = m \sum_{e \in F_1} \mathrm{FVO}(e,\mathrm{RI}).$$

Here α is m^2 times the average distance between RI and any face in F_1. Thus the larger the number of vectors in F_1, the stronger is the impact. Similarly, we define β the counterpart of α at the opposite side of the threshold to be:

$$\beta = n\left(\sum_{e\in F_2} \mathrm{FVO}(e,\mathrm{CIF}) + \sum_{e\in F_3} \mathrm{FVO}(e,\mathrm{RI})\right).$$

Finally, we obtain the adjusted face verification output that accounts for the number of hits recorded in the cross-matching phase. The value we generate falls in the same interval of the FVO, and the same OCT is used for the decision.

$$\mathrm{CMO}(\mathrm{CIF},\mathrm{RI}) = \frac{\alpha+\beta+\omega\times \mathrm{FVO}(\mathrm{CIF},\mathrm{RI})}{\omega+m^2+n^2}.$$

If we reconsider the example in Figure 15, assuming that the initial classification value $\mathrm{FVO}(\mathrm{CIF},\mathrm{RI})$ is 1.46, and the initial weight attributed to that value is $\omega = 100$, the output of the algorithm should be:

$F_1 = \{\mathrm{face}_i, \mathrm{face}_j, \mathrm{face}_k\}.$

$F_2 = \{\mathrm{face}_l, \mathrm{face}_m\}.$

$F_3 = \{\mathrm{face}_m, \mathrm{face}_n\}.$

$m = |F_1| = 3.$

$n = |F_2| + |F_3| = 4.$

$\alpha = 3 \times \left(\mathrm{FVO}(\mathrm{face}_i,\mathrm{RI}) + \mathrm{FVO}\left(\mathrm{face}_j,\mathrm{RI}\right) + \mathrm{FVO}(\mathrm{face}_k,\mathrm{RI})\right),$

$\alpha = 3 \times (.42 + .37 + .56) = 3 \times 1.35 = 4.05.$

$\beta = 4 \times \left(\mathrm{FVO}(\mathrm{face}_l,\mathrm{RI}) + \mathrm{FVO}(\mathrm{face}_m,\mathrm{RI}) + \mathrm{FVO}(\mathrm{face}_n,\mathrm{RI}) + \mathrm{FVO}(\mathrm{face}_o,\mathrm{RI})\right),$

$\beta = 4 \times (1.89 + 2.02 + 1.91 + 2.33) = 4 \times 8.15 = 32.06.$

$\mathrm{CMO}(\mathrm{CIF},\mathrm{RI}) = \frac{4.05+32.06+100\times 1.46}{100+3^2+4^2} = \frac{4.05+32.06+146}{100+9+16} = \frac{182.11}{125} = 1.46.$

The final decision after running cross-matching is not different than the initial output. It can be considered as no improvement in case of an incorrect initial decision, or good conclusion in case it consolidates the expectation. The fact that the value of the decision is not heavily affected is due to the contradictory list of force vectors generated

by the cross-matching mechanism. In fact, such unusual cases are rare, and the dominant observed results are not contradictory. In the example above, we try to show all types of vectors and simulate the algorithm's process regardless of the meaningfulness of the example. Chapter IV covers more concrete cases.

3. Experiment 3: Weighted Cross-Matching

Similar to experiment 2, we use the same concept of cross-matching but we introduce the notion of weights to attribute to the force vectors. We describe the differences between the two methods used in experiment two and three and explain the purpose of additional weights by highlighting the role of metadata.

a. Algorithm

We reconsider the results of the previous experiment where α and β are computed without discrimination; all the observations have the same weight of "one." However, some comparisons show more relevant connection to the CIF than others based on how many elements of the image metadata they share. Therefore, we assign high weight to the FVO(TCI, CIF) if the two inspected faces share context such as the same location or the same timeframe. In addition, in experiment two, we only isolate the pairwise comparisons that show a high level of confidence (FVO < LCL). At this stage of cross-matching, we account for partially obscured faces. Therefore, when comparing the CIF to the TCI, we include the cases where FVO(TCI, CIF) ≤ OCT instead of the limited range of FVO(TCI, CIF) ≤ LCL used in the previous experiment. Alternatively, the initial weight attributed to the FVO is relatively low but increases with the number of shared metadata fields. Nevertheless, LCL and UCL are still used for determining the CFS and DFS.

In the implementation, we use the EXIF (Exchangeable Image File Format) tool to access the metadata. EXIF is a standard for metadata format for images issued from digital cameras. It includes a large set of information such as the flash (on/off) and exposure time, in addition to contextual data like the timestamp and GPS tag. The list of fields provided by EXIF is large, but we limit our focus on a short list we use to create a set of variables that we enumerate in Table 4 and that we use during the comparison of

two faces to determine the degree of similarity. The left fields are either correlated to a selected one or irrelevant and cannot infer the context/content of the image.

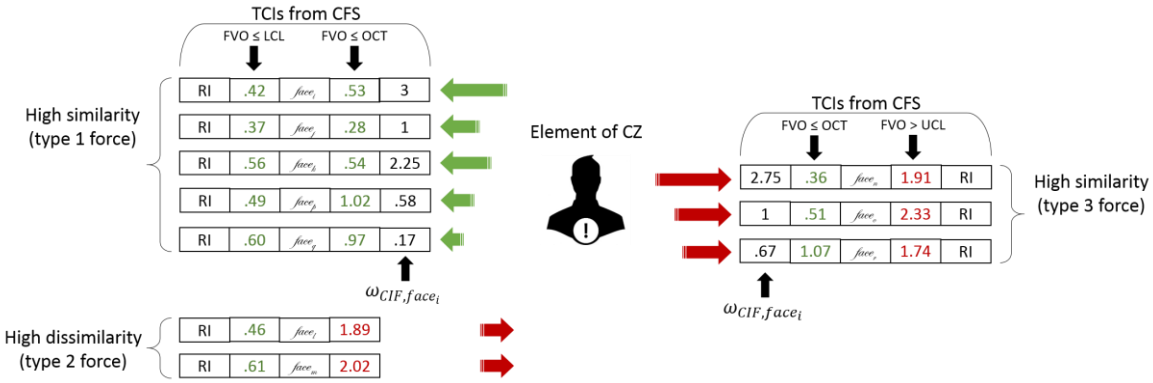
Table 4. List of Variables Derived from the Metadata

Variables	Class / Semantic
image_path	Categorical (binary) One if the two faces (the CIF and the TCI) are in images belonging to the same folder. Zero otherwise.
diff_nbr_faces	Numerical (positive integer) The difference in number of faces present on the images containing the CIF and TCI.
diff_filesize	Numerical (positive integer) The difference in bytes of the sizes of the images containing the CIF and TCI. (It infers same content when near zero.)
diff_timestamp	Numerical (positive integer) The elapsed time in seconds between the instant of capture of the two images.
Distance	Numerical (positive integer) The distance in meters between the two GPS locations of capture of the two images. Missing values.
same_camera	Categorical (binary) One if the two images were captured with the same camera (make/model). Zero otherwise.

In order to say which TCI is more relevant than another, we need to measure how close two faces are to each other, in the sense of context and not content. For example, we need to set limits for distance, time, and file size. This is to answer the question *How near is near?* Based on the thresholds, we can evaluate the similarity between two faces referring to their images' metadata. The thresholds we use are as follow:

- Distance threshold: 150 meters.
- Timeframe threshold: 5 minutes.
- File size threshold: 200 KB.

The derived weights affect the force vectors of type one and type three but not type two vectors since they represent repulsions. Thus, a similarity or dissimilarity in the metadata cannot infer any conclusion. Moreover, higher importance is attributed to the distance and the timeframe as they are the most predictive variables for the classification. This notion of weights is introduced to account for the trustworthiness of the TCIs and to take into consideration how representative a TCI can be to both the RI and the CIF. In short, we combine cross-matching as defined in experiment two with the metadata and check whether the global success rate improves or not. Figure 16 illustrates the impact of metadata by attributing weights to the preexisting vectors in Figure 15. We also note the presence of new vectors that are added accounting for flexible thresholds when comparing the CIF to the TCI considering partial obscuration. In keeping with [21], we seek to implement a framework that considers spatial and temporal re-occurrence, without involving machine-learning techniques.



The comparisons with shared context have high impact, whereas when less metadata fields match, the force vectors have significantly less attraction.

Figure 16. Weighted Cross-Matching, Metadata Impact.

The revised mathematical formula of the algorithm defined in experiment two follows. This new version addresses the weights attributed to the force vectors generated after the cross-matching process. We also define another parameter called λ the weight adjustment factor. We set λ to be equal to the number of variables (from Table 4) but attributing higher weight to the distance and the timeframe.

$$\lambda = 1\left(\text{same}_{\text{path}}\right) + 1\ (\#\text{faces}) + 1\ (\text{same}_{\text{camera}}) + 1\ (\text{file}_{\text{size}} \leq 200\text{KB}) +$$
$$4\ (\text{distance} \leq 150m) + 4\ (\text{timeframe} \leq 5') \rightarrow \lambda = 12.$$

$\omega_{x,y}$ is the adjusted weight applied to the vector of attraction force based on the face verification output, the weight adjustment factor, and the number of metadata fields shared by face x and face y. $\omega_{x,y}$ is computed by applying to the algorithm described in Figure 17 which represents the case of comparing the CIF to a TCI from the CFS.

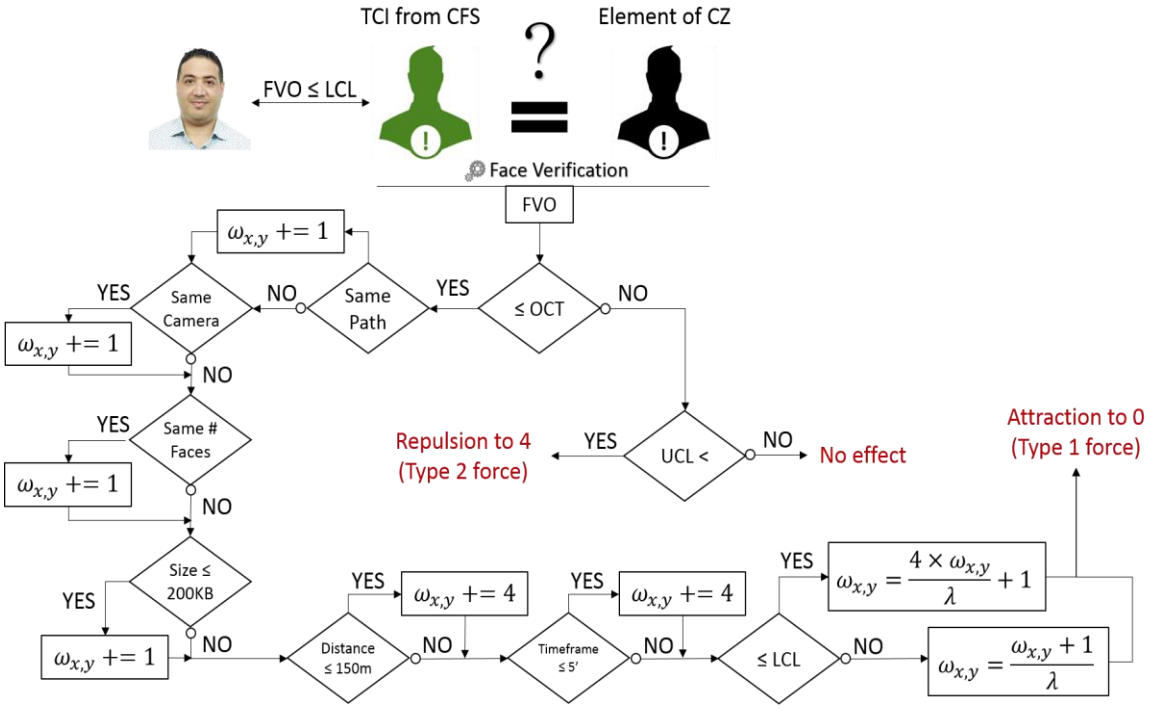


Figure 17. Computing the Adjusted Weight.

The sum of weights for the F_1 set, which was simply the total number of vectors, is now replaced by the total of the adjusted weights.

$$m = \sum_{e \in F_1} \omega_{\text{CIF},e}.$$

In the same fashion, the sum of weights for both F_2 and F_3 is adjusted to be

$$n = \sum_{e \in F_2, F_3} \omega_{\text{CIF},e}.$$

Consequently, the new α and β values are computed using the new m and n values in addition to applying the weight adjustment to each pairwise face verification output to give:

$$\alpha = m\sum_{e\in F_1}\omega_{\text{CIF},e}\times \text{FVO}(e,\text{RI}),$$

$$\beta = n\left(\sum_{e\in F_2}\omega_{\text{CIF},e}\times \text{FVO}(e,\text{CIF}) + \sum_{e\in F_3}\omega_{\text{CIF},e}\times \text{FVO}(e,\text{RI})\right).$$

The final step to obtain the revised face verification output remains unchanged but uses the new generated parameters,

$$\text{WCM(CIF, RI)} = \frac{\alpha+\beta+\omega\times\text{FVO(CIF,RI)}}{\omega+m^2+n^2}.$$

As an illustration, we reconsider the example in Figure 15 by applying the results in Figure 16. Now:

$\lambda = 12.$

$\omega_{\text{CIF,face}_i} = 3 \qquad \omega_{\text{CIF,face}_j} = 1 \qquad \omega_{\text{CIF,face}_k} = 2.25,$

$\omega_{\text{CIF,face}_p} = .58 \qquad \omega_{\text{CIF,face}_q} = .17,$

$\omega_{\text{CIF,face}_l} = 1 \qquad \omega_{\text{CIF,face}_m} = 1,$

$\omega_{\text{CIF,face}_n} = 2.75 \qquad \omega_{\text{CIF,face}_o} = 1 \qquad \omega_{\text{CIF,face}_r} = .67.$

$m = 3 + 1 + 2.25 + .58 + .17 = 7.$

$n = 1 + 1 + 2.75 + 1 + .67 = 6.42.$

$$\alpha = m\times\Big(\omega_{\text{CIF,face}_i}\times\text{FVO}(\text{face}_i,\text{RI}) + \omega_{\text{CIF,face}_j}\times\text{FVO}\big(\text{face}_j,\text{RI}\big) + \omega_{\text{CIF,face}_k}\times \text{FVO}(\text{face}_k,\text{RI}) + \omega_{\text{CIF,face}_p}\times\text{FVO}\big(\text{face}_p,\text{RI}\big) + \omega_{\text{CIF,face}_q}\times\text{FVO}\big(\text{face}_q,\text{RI}\big)\Big),$$

$$\alpha = 7\times(3\times.42 + 1\times.37 + 2.25\times.56 + .58\times.49 + .17\times.6),$$

$$\alpha = 7\times(1.26 + .37 + 1.26 + .28 + .1) = 7\times 3.27 = 22.89.$$

$$\beta = n\times\Big(\omega_{\text{CIF,face}_l}\times\text{FVO}(\text{face}_l,\text{CIF}) + \omega_{\text{CIF,face}_m}\times\text{FVO}(\text{face}_m,\text{CIF}) + \omega_{\text{CIF,face}_n}\times \text{FVO}(\text{face}_n,\text{RI}) + \omega_{\text{CIF,face}_o}\times\text{FVO}(\text{face}_o,\text{RI}) + \omega_{\text{CIF,face}_r}\times\text{FVO}(\text{face}_r,\text{RI})\Big),$$

$$\beta = 6.42\times(1\times 1.89 + 1\times 2.02 + 2.75\times 1.91 + 1\times 2.33 + .67\times 1.74),$$

$$\beta = 6.42\times(1.89 + 2.02 + 5.25 + 2.33 + 1.17) = 6.42\times 12.66 = 81.28.$$

$$\text{WCM(CIF, RI)} = \frac{22.89+81.28+100\times 1.46}{100+7^2+6.42^2} = \frac{22.89+81.28+146}{100+49+41.22} = \frac{250.17}{190.22},$$

$$\text{WCM(CIF, RI)} = 1.32.$$

The final decision of the classifier is now much different than the output of both the first and the second experiments. This intended change has an effect on the global success rate of the classification. We note that the output 1.32 is lower than the initial value 1.46, although we have the same number of type-one force vectors and type-two and type-three force vectors. The decrease in the output in meaningful. $^2/_5$ of the elements in the F_1 set are relevant and exert strong attraction, while in $F_2 \cup F_3$ only $^1/_5$ is showing effect. Therefore, the final attraction applied on the decision is toward class one.

b. Threshold Determination

Varying the parameters of the algorithm and the thresholds affects the performance of the framework in terms of accuracy and execution time. Therefore, we investigate the different methods to obtain the appropriate set of values while keeping the system fully automated and returning satisfactory results.

(1) Methodology

Three of our selected variables require thresholds: distance, timeframe, and file size. We choose these thresholds by holding two thresholds fixed and varying the third. After executing multiple tests, we pick the threshold values that achieve the highest performance.

The thresholds that we define and use in the third experiment are not optimal but are good enough to evaluate the performance of the algorithm. Determining the best values for the thresholds is challenging due to the complexity of the algorithm $O(n^2)$. Although we use a Hadoop cluster and MapReduce for fast processing, every evaluation lasts around 28 minutes. Moreover, we opt for satisfactory thresholds instead of optimal thresholds to account for the future generalization of the method. In fact, the thresholds we use are coupled to dataset, whereas we expect the algorithm to demonstrate improvement if deployed on other platforms and running on a different image corpus. Therefore, introducing acceptable margin of flexibility avoids being sensitive to the dataset. Otherwise, we end up overfitting.

(2) K-Nearest Neighbors and Distributed Random Forest Attempts

In this section, we describe a statistical approach to choose thresholds, which for a number of reasons described in this section failed. Here, we generate a dataset to include only the pairwise face verification output between the CIF and the TCI with the metadata but without involving the RI. As a response, we write one if CIF and the TCI are the same person and zero otherwise. Our goal here is to create a way to determine if two faces are neighbors or not, to be used for clustering and voting. This initial approach using either K-nearest neighbors or distributed random forests fails to pass the validation phase. The conclusion is that we cannot combine the metadata with the FVO to measure the distance between two faces, no matter the algorithm used for training. This is due to the large number of outliers and contradictory observations. We witness records with low FVO and shared metadata with one as response (which was expected), whereas many other observations that do not have common metadata values but low FVO have also one for response. It gets more complicated when elements share multiple metadata fields but high FVO and still obtain one as response. These three illustrations are not unusual given the nature of the dataset, which derives from an unconstrained PDIL. Another factor that contributes heavily to the failure of the K-nearest neighbors approach is the large discrepancy between the number of true positives and true negatives per profile. Training the data leads to under-fitting, and the model tends to have high error rates on class one. Table 5 shows the error rates when fitting a distributed random forest model on a sample of 385,980 comparisons.

Table 5. Distributed Random Forest and Class Balance Impact.

Class Balance	Set	Count	Class	Count	Prediction class		Error rate
					0	1	
Applied	Training	308,784	0	290,133	281,429	8,704	.03
			1	18,651	8,764	9,887	.47
	Test	77,196	0	72,481	70,306	2,175	.03
			1	4,715	2,169	2,546	.46
Not Applied	Training	308,784	0	290,133	281,211	8,922	.03
			1	18,651	1,313	17,338	.07
	Test	77,196	0	72,481	68,141	4,340	.06
			1	4,715	2,146	2,569	.46

By applying class balancing, we over-fit the class one. The error rate for class zero remains relatively low, but the impact of the minor increase in that error is drastic given the total number of observations in that class. The model is not reliable and requires additional analysis that might require revisiting the response construction. We might also be missing predictors or we probably need to drop high-leverage observations.

In a second attempt, we try to determine the thresholds that properly reflect the links between images (comparing metadata), accounting for the interaction between variables, to generate a probability that can be used as a trustworthiness evaluator. Therefore, we use the output of experiment two with all the predictors, including the metadata, to help the classifier assign weights to the pairwise comparisons before averaging. The model we obtain is trained to determine if the current face should be attracted toward class one, which means the same person, or toward class zero, the opposite, for a given pairwise comparison. Based on that decision, if it complies with the type of force to be applied, an extra weight is assigned to that observation. In case of contradiction, we lower the impact of that comparison on the center of mass by dividing the FVO by a given weight and penalize the observation because of non-resemblance.

To train the algorithm, we use the distributed random forest algorithm for a variety of reasons. First, since random forest inherits all the properties of trees, it naturally handles interaction between variables, which is the main idea behind adding metadata to the model. Another contribution of distributed random forest is that it is resistant to outliers, and in our dataset we have many odd observations. We do not need to perform variable selection. Nonetheless, we discard many fields of the metadata due to non-context relevance or correlation. Distributed random forests do not require any transformation of the values. This could be an issue with the numeric values when several values are of high magnitude, while others do not exceed 10. Finally, distributed random forests do well with missing values, and here we have a great portion of rows without GPS information. These missing values are not necessarily due to the absence of the GPS capability on the device, but more because of an authorization matter.

In this experiment, we address the functions applied to the FVO–center of mass and square function–and attempt to compute the decision using a statistical approach. The

number of pairwise comparisons generated from cross-matching is 1,404,864,989, where class one has 35,735,303 records (2.54%). and class 0 has 1,369,129,686 observations (97.46%). Performing machine-learning on such a huge dataset is non-trivial, and sampling has to take place. For that reason, we randomly select 1% of the total records and split it into three subsets: 60% training set, 20% validation set, and 20% test set. This process is made only one time and is applied to all subjects/profiles without creating per-user patterns. Table 6 describes variables and the response in the model we use for training and prediction.

Table 6. List of Variables in the Distributed Random Forest Model.

List of variables	Class / Semantic
FVO(CIF, RI)	Numerical (float between zero and four). The output of face verification when comparing the CIF to the RI.
FVO(CIF, TCI)	Numerical (float between zero and four). The FVO when comparing the CIF with a trusted classified image from either side of the threshold.
FVO(TCI, RI)	Numerical (float between zero and four). The FVO when comparing the TCI with the RI of the UFF.
same_path	Categorical
diff_nbr_faces	Numerical
diff_filesize	Numerical
diff_timestamp	Numerical
Distance	Numerical
same_camera	Categorical
response	Categorical (binary) One if the CIF is the UFF. Zero otherwise.

The same anomalies observed in the K-nearest neighbors attempt are also spotted in this model, although we note minor improvement. We cannot rely on the predictions of this model and decide to reject this approach of using statistical model for determining the relevance of a TCI against the RI and the CIF.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS AND ANALYSIS

A. EXPERIMENT 1: FACE VERIFICATION

In this experiment, we run the OpenFace classifier 15 times for 15 different profiles on 27,115 labeled faces. At this stage, we record the results to use as reference for future comparisons with the upcoming experiments and determine the appropriate thresholds to use in the next phases.

Different techniques exist for threshold selection, and there is no unique solution. The threshold selection relies on the users' objectives and how tolerant they are to false positives and false negatives. In this section, we present five measures to evaluate threshold choice and select the one that suits our needs. All these measures are evaluated separately on all the profiles–classification of 27,115 faces per subject–and then summarized in a single report gathering all 15 results as one output. The reason to sum all the profiles' results is that the individual outputs can be misleading as they are heavily affected by the number of true positives, which varies from one profile to another. For instance, the threshold that gives the best success rate can classify all faces as being different from the UFF for a profile that has only a few images, whereas for someone having more than 10% of the corpus size, the success rate imposes a higher threshold because the number of true positives can no longer be neglected. These terms are used in this chapter:

- TP: true positive.
- TN: true negative.
- FP: false positive.
- FN: false negative.
- TNR: true negative rate $=$ Specificity $= \frac{TN}{TN + FP}$.
- FPR: false positive rate $= \frac{FP}{TN + FP} = 1 -$ Specificity.
- TPR: true positive rate $=$ Sensitivity $=$ Recall $= \frac{TP}{TP + FN}$.
- Precision $= \frac{TP}{TP + FP}$.
- Success Rate $=$ Accuracy $= \frac{TP + TN}{TP + FN + TN + FP} = 1 -$ Misclassification.

The five threshold selection criteria that we look at are

- F1 score (also called f-score, or f-measure): $2 \times \frac{Precision \times Recall}{Precision + Recall}$.

  Figure 18 shows the F1 score distribution grouping all the profiles. We pick the threshold that maximizes the F-score.

- Equal Positive Rate point (EPR): The threshold where TPR = FPR. It is the intersection of the Receiver Operating Characteristic curve (ROC) with the diagonal line from the top left corner (perfect classification point) to the bottom right corner. In Figure 19, the cyan plot point is the intersection between the ROC curve and the diagonal. The threshold for that intersection is 1.10.

- Nearest Threshold on the ROC to the Perfect classification point (NTP). Figure 19 illustrates the global NTP with the magenta plot: NTP = 1.08.

- Equal Success/Error Rate point (ESR): The threshold where the specificity is equal to the sensitivity. Figure 20 shows the intersection between the success rate and the error rate curves. The corresponding threshold is 1.09.

- Best Success Rate (BSR): The threshold that maximizes the accuracy. The BSR in Figure 20 is 0.5.

- Histogram Intersection Point (HIP): The intersection of the histogram of true positives and the histogram of true negatives. A normalization must take place to account for the inter-classes unbalance. Figure 21 shows both histograms and their intersection: 1.01.

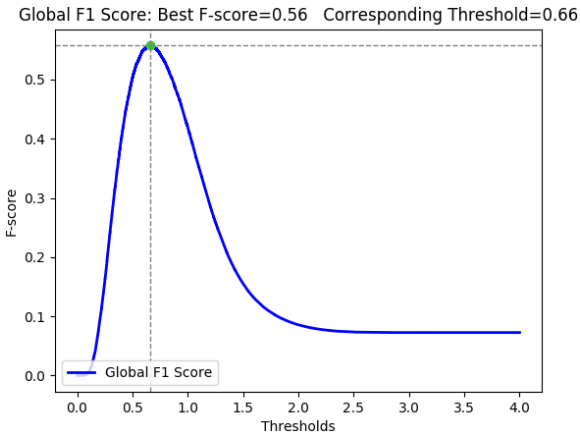


Figure 18. Global F1 Score.

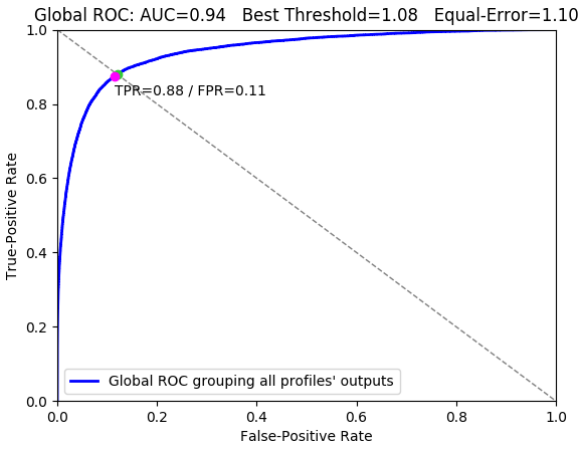


Figure 19. ROC Curve Grouping 15 Classifications.

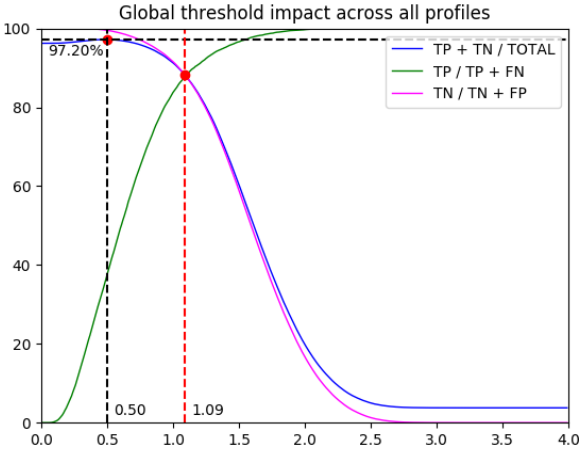


Figure 20. Sensitivity/Specificity across all Profiles.

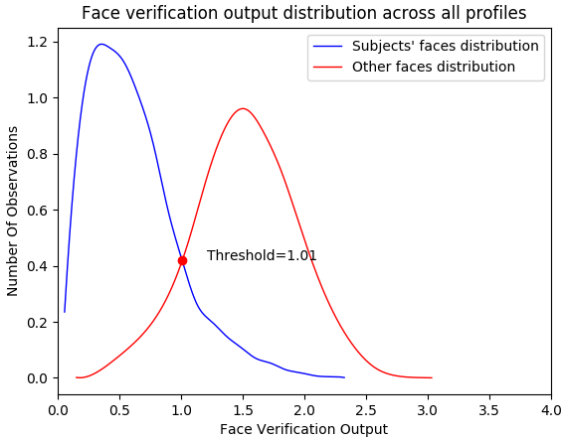


Figure 21. Normalized Threshold Distribution per Class.

Considering the different proposed thresholds, we start by ignoring the F-score and the BST because they are not representative of the data and yield much lower thresholds (much lower than the recommended threshold by [12] of 1.0). Next, we discard the HIP due to its sensitivity to normalization. The NTP, EPR, and BST are very similar; however, we do not prioritize either the true positives or the true negatives. Thus, we are indifferent to their respective success/error rate. (They do not have to be equal.) Therefore, we opt for the NTP = 1.08, principally because it is resistant to the interclass unbalance and, by definition, the NTP is the closest threshold to the perfect classification point rather than the EPR that suggests that the true positive and negative rates are the same. Another key point is that the NTP is not heavily affected by the cross-profile variance, as shown in Table 7.

Table 7. Different Thresholds across all the Profiles.

Profiles	Faces Count	ROC curve			Success/Error rates curves			HIP		F1 Score	
		AUC	NTP	EPR	ESR	BSR	Succ.	Stand.	Norm.	F-score	Threshold
Abrha	667	0.87	1.14	1.18	1.17	0.60	97.68%	0.49	1.06	0.37	0.74
Astraat	26	0.99	0.94	0.94	0.93	0.50	99.95%	0.44	0.91	0.7	0.49
Bonji	70	0.93	1.05	1.17	1.17	0.55	99.81%	0.48	0.95	0.49	0.67
Heeda	33	0.82	1.30	1.22	1.21	-	99.88%	-	1.07	0.05	0.26
Hickam	88	0.98	1.23	1.22	1.21	0.45	99.68%	0.37	1.15	0.43	0.76
Hmouda	5	1.00	0.46	0.46	0.46	-	99.98%	-	0.45	0.29	0.31
Holu	19	0.99	1.06	1.27	1.26	0.63	99.95%	0.44	1.01	0.62	0.66
Khufu	2,227	0.97	1.27	1.29	1.28	0.93	96.67%	0.86	1.19	0.79	1
Laghbesh	14	0.97	1.13	1.13	1.13	0.36	99.96%	0.35	0.91	0.48	0.45
Mekah	179	0.96	1.19	1.28	1.27	0.59	99.62%	0.60	1.10	0.69	0.73
Mimyth	1,074	0.96	1.19	1.20	1.19	0.83	97.35%	0.70	1.10	0.66	0.93
Sakis	29	0.87	0.98	1.01	1.00	-	99.89%	-	0.87	0.11	0.32
Sierra	3,412	0.92	1.09	1.11	1.11	0.72	93.51%	0.65	0.98	0.71	0.81
Sokhoi	2,634	0.90	0.83	0.84	0.84	0.49	93.87%	0.40	0.77	0.62	0.55
Yakouza	4,836	0.95	0.86	0.88	0.88	0.59	91.93%	0.47	0.80	0.76	0.66
Global	15,313	0.94	**1.08**	1.1	1.09	0.50	97.20%	0.37	1.01	0.56	0.66

Now we select the values for LCL and UCL to be used in the next experiments. One suggestion is that we set the LCL and UCL to be the thresholds where we are 95% confident that the face is a true positive or a true negative. The LCL is the value where the ratio of true positives $\frac{TP}{TP+FP+TN+FN}$ drops below .95 when varying the threshold. Similarly, the UCL is the value for which the ratio of true negatives $\frac{TN}{TP+FP+TN+FN}$ exceeds .95. Figure 22 shows how such values cannot be used due to the steep discrepancy between the cardinality of class one and class zero.

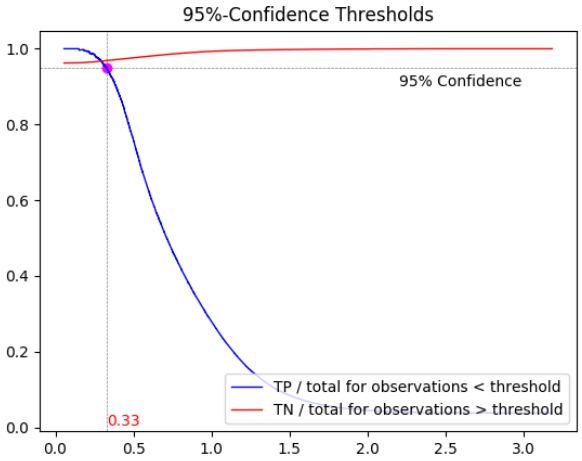


Figure 22. LCL and UCL with 95% Confidence Approach.

In fact, in the graph in Figure 22, grouping all 15 classifications, the ratio of the true positives is only:

$$\frac{15\ 313}{15 \times 27115} = 3.76\%$$

In other words, we are always certain that a face is of class zero with more than 95% confidence. (The probability is 0.96.) Another approach must be considered. We re-examine the normalized histograms and decide to spot the thresholds that split both histograms, true positive and negative histograms, in the middle. In that case, the LCL is the median of the true positive histogram. Thus, any value below that threshold is very likely to be a true positive. Consequently, the UCL is the median of the true negatives.

Figure 23 shows the values of the final LCL and UCL that are used in the cross-matching of the next experiments. Table 8 shows the variance of the UCL and LCL across the profiles.

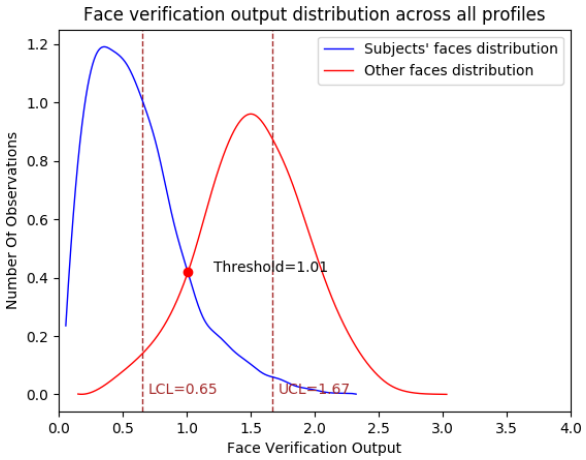


Figure 23. LCL and UCL with Median Approach.

Table 8. UCL and UCL per Profile.

Profile	UCL	LCL
Abrha	0.89	1.46
Astraat	0.5	1.65
Bonji	0.77	1.51
Heeda	0.98	1.63
Hickam	0.81	1.81
Hmouda	0.4	1.77
Holu	0.65	1.98
Khufu	0.85	1.86
Laghbesh	0.64	1.51
Mekah	0.69	1.66
Mimyth	0.83	1.76
Sakis	0.57	1.55
Sierra	0.68	1.55
Sokhoi	0.53	1.39
Yakouza	0.51	1.54
Global	**0.65**	**1.67**

In Table 9, we record the results we use as benchmarks in the next experiments. We also introduce the Balanced Inter-class Accuracy (BIA). Since the number of elements of class zero are huge (for most of the profiles) compared to the cardinality of class one, the simple use of traditional measures of accuracy cannot provide good results. In fact, the average number of true positives per profile is 1,021, which represents 3.76% of the corpus size. A simple attribution of class zero to all the observations raises the accuracy to 96.24%, thus better than the accuracy generated by the FV algorithm. Therefore, we average the per class misclassification rate and derive a balanced inter-class accuracy as shown in Table 9.

Table 9. Final Summary of the Face Verification Process.

Profile	Total Count	True Positives		True Negatives		$\frac{TP+TN}{Total}$	BIA
		Detected	Total	Detected	Total		
Abrha	27,115	483	667	22,783	26,448	85.80%	79.28%
Astraat	27,115	26	26	24,649	27,089	91.00%	95.50%
Bonji	27,115	61	70	25,255	27,045	93.37%	90.26%
Heeda	27,115	22	33	22,271	27,082	82.22%	74.45%
Hickam	27,115	76	88	25,968	27,027	96.08%	91.22%
Hmouda	27,115	5	5	24,557	27,110	90.58%	95.29%
Holu	27,115	18	19	26,589	27,096	98.13%	96.43%
Khufu	27,115	1,858	2,227	24,132	24,888	95.85%	90.20%
Laghbesh	27,115	12	14	25,141	27,101	92.76%	89.24%
Mekah	27,115	155	179	26,221	26,936	97.27%	91.97%
Mimyth	27,115	925	1074	24,801	26,041	94.88%	90.68%
Sakis	27,115	24	29	20,253	27,086	74.78%	78.77%
Sierra	27,115	2,830	3,412	20,509	23,703	86.07%	84.73%
Sokhoi	27,115	2,388	2,634	16,210	24,481	68.59%	78.44%
Yakouza	27,115	4,498	4,836	17,539	22,279	81.27%	85.87%
Global	**406,725**	**13,381**	**15,313**	**346,878**	**391,412**	**88.58%**	**88.00%**

In summary, experiment one helps record the performance of the face verification classifier to be used as reference in the next experiments, in addition to determining the required parameters.

Below are the settings we use in the next experiments:

- OCT = 1.08
- LCL = 0.65
- UCL = 1.67
- Global Success Rate (GSR) = 88.58%
- Global Balanced Interclass Accuracy = 88.00%

B. EXPERIMENT 2: CROSS-MATCHING

The idea in this experiment is to reevaluate the elements of the CZ using TCIs and compare the output to the report generated in the first experiment. First, in Table 10, we show the number of TCIs per profile and their classification.

Table 10. TCIs Distribution per Profile.

Profiles	CFS		DFS	
	Class 0 (FP)	Class 1 (TP)	Class 0 (TN)	Class 1 (FN)
Abrha	239		8,071	
	110	129	8,043	28
Astraat	83		11,739	
	64	19	11,739	-
Bonji	50		6,798	
	23	27	6,797	1
Heeda	1,186		11,855	
	1,180	6	11,852	3
Hickam	58		16,954	
	31	27	16,954	-
Hmouda	528		13,040	
	523	5	13,040	-
Holu	19		20,130	
	8	11	20,130	-

Table 10. TCIs Distribution per Profile (Cont.).

Profiles	CFS		DFS	
	Class 0 (FP)	Class 1 (TP)	Class 0 (TN)	Class 1 (FN)
Khufu	717		15,713	
	26	691	15,657	56
Laghbesh	39		7,369	
	31	8	7,369	-
Mekah	107		12,878	
	17	90	12,874	4
Mimyth	276		12,085	
	40	236	12,068	17
Sakis	1,598		8,592	
	1,581	17	8,591	1
Sierra	1,813		8,929	
	150	1,663	8,850	79
Sokhoi	3,309		5,307	
	1,556	1,753	5,275	32
Yakouza	4,828		7,838	
	1,140	3,688	7,793	45

Table 11 presents a summary of the results for both experiment one and experiment two, highlighting the per-class error rate, the per-profile accuracy, the per-profile balanced inter-class accuracy, and the global outputs aligning all the profiles together. We note that the accuracy and the BIA do not evolve in the same direction per profile. Nevertheless, we register a satisfying overall improvement on both parameters, accuracy and BIA.

Table 11. Summary of Experiment 2.

Profiles	Experiment 1				Experiment 2			
Abrha	TP	FN	TN	FP	TP	FN	TN	FP
	483	184	22,783	3,665	472	195	24,186	2,262
Error rate	27.59%		13.86%		29.24%		8.55%	
F-Score	**0.20062**				↑ **0.27757**			
Accuracy	**85.80%**				↑ **90.94%**			
BIA	**79.28%**				↑ **81.11%**			
Astraat	TP	FN	TN	FP	TP	FN	TN	FP
	26	0	24,649	2,440	25	1	25,197	1,892
Error rate	0.00%		9.01%		3.85%		6.98%	
F-Score	**0.02087**				↑ **0.02573**			
Accuracy	**91.00%**				↑ **93.02%**			
BIA	**95.50%**				↓ **94.58%**			
Bonji	TP	FN	TN	FP	TP	FN	TN	FP
	61	9	25,255	1,790	59	11	26,588	457
Error rate	12.86%		6.62%		15.71%		1.69%	
F-Score	**0.06351**				↑ **0.20137**			
Accuracy	**93.37%**				↑ **98.27%**			
BIA	**90.26%**				↑ **91.30%**			
Heeda	TP	FN	TN	FP	TP	FN	TN	FP
	22	11	22,271	4,811	20	13	22,438	4,644
Error rate	33.33%		17.76%		39.39%		17.15%	
F-Score	**0.00904**				↓ **0.00852**			
Accuracy	**82.22%**				↑ **82.83%**			
BIA	**74.45%**				↓ **71.73%**			
Hickam	TP	FN	TN	FP	TP	FN	TN	FP
	76	12	25,968	1,059	74	14	26,512	515
Error rate	13.64%		3.92%		15.91%		1.91%	
F-Score	**0.12428**				↑ **0.21861**			
Accuracy	**96.05%**				↑ **98.05%**			
BIA	**91.22%**				↓ **91.09%**			
Hmouda	TP	FN	TN	FP	TP	FN	TN	FP
	5	0	24,557	2,553	5	0	24,559	2,551
Error rate	0.00%		9.42%		0.00%		9.41%	
F-Score	**0.00390**				↑ **0.00390**			
Accuracy	**90.58%**				↑ **90.59%**			
BIA	**95.29%**				↑ **95.30%**			

Table 11. Summary of Experiment 2 (Cont.).

Profiles	Experiment 1				Experiment 2			
Holu	TP	FN	TN	FP	TP	FN	TN	FP
	18	1	26,589	507	18	1	26,907	189
Error rate	5.26%		1.87%		5.26%		0.70%	
F-Score	**0.06618**				↑ **0.15929**			
Accuracy	**98.13%**				↑ **99.30%**			
BIA	**96.43%**				↑ **97.02%**			
Khufu	TP	FN	TN	FP	TP	FN	TN	FP
	1,858	369	24,132	756	2,000	227	24,363	525
Error rate	16.57%		3.04%		10.19%		2.11%	
F-Score	**0.76761**				↑ **0.84175**			
Accuracy	**95.85%**				↑ **97.23%**			
BIA	**90.20%**				↑ **93.85%**			
Laghbesh	TP	FN	TN	FP	TP	FN	TN	FP
	12	2	25,141	1,960	12	2	26,513	588
Error rate	14.29%		7.23%		14.29%		2.17%	
F-Score	**0.01208**				↑ **0.03909**			
Accuracy	**92.76%**				↑ **97.82%**			
BIA	**89.24%**				↑ **91.77%**			
Mekah	TP	FN	TN	FP	TP	FN	TN	FP
	155	24	26,221	715	153	26	26,686	250
Error rate	13.41%		2.65%		14.53%		0.93%	
F-Score	**0.29552**				↑ **0.52577**			
Accuracy	**97.27%**				↑ **98.98%**			
BIA	**91.97%**				↑ **92.27%**			
Mimyth	TP	FN	TN	FP	TP	FN	TN	FP
	925	149	24,801	1,240	960	114	25,269	772
Error rate	13.87%		4.76%		10.61%		2.96%	
F-Score	**0.57116**				↑ **0.68425**			
Accuracy	**94.88%**				↑ **96.73%**			
BIA	**90.68%**				↑ **93.21%**			
Sakis	TP	FN	TN	FP	TP	FN	TN	FP
	24	5	20,253	6,833	23	6	21,083	6,003
Error rate	17.24%		25.23%		20.69%		22.16%	
F-Score	**0.00697**				↑ **0.00760**			
Accuracy	**74.48%**				↑ **77.84%**			
BIA	**78.77%**				↓ **78.57%**			

Table 11. Summary of Experiment 2 (Cont.).

Profiles	Experiment 1				Experiment 2			
Sierra	TP	FN	TN	FP	TP	FN	TN	FP
	2,830	582	20,509	3,194	2,928	484	21,049	2,654
Error rate	17.06%		13.48%		14.19%		11.20%	
F-Score	**0.59983**				↑ **0.65110**			
Accuracy	**86.07%**				↑ **88.43%**			
BIA	**84.73%**				↑ **87.31%**			
Sokhoi	TP	FN	TN	FP	TP	FN	TN	FP
	2,388	246	16,210	8,271	2,322	312	16,548	7,933
Error rate	9.34%		33.79%		11.85%		32.40%	
F-Score	**0.35929**				↑ **0.36031**			
Accuracy	**68.59%**				↑ **69.59%**			
BIA	**78.44%**				↓ **77.88%**			
Yakouza	TP	FN	TN	FP	TP	FN	TN	FP
	4,498	338	17,539	4,740	4,543	293	17,370	4,909
Error rate	6.99%		21.28%		6.06%		22.03%	
F-Score	**0.63919**				↓ **0.63592**			
Accuracy	**81.27%**				↓ **80.82%**			
BIA	**85.87%**				↑ **85.95%**			
Global	TP	FN	TN	FP	TP	FN	TN	FP
	13,381	1,932	346,878	44,534	13,614	1,699	355,268	36,144
Error rate	12.62%		11.38%		11.10%		9.23%	
F-Score	**0.36546**				↑ **0.41844**			
Accuracy	**88.58%**				↑ **90.70%**			
BIA	**88.00%**				↑ **89.84%**			

In essence, cross-matching shows a decrease in the overall misclassification rate of the face verification, reducing it from 11.42% to 9.3%. The global BIA reflects an improvement that can be confirmed by looking at the global per class error rate, which shows a decrease for both classes of the classifier. We note that the impact is affecting the false positives more than all others. Nevertheless, that portion of the data is relatively large, and reducing its cardinality leads to a refined subset of faces for deeper analysis using more complex techniques.

Contrary to experiment one, the number of observations per profile seems to have great influence on the amount of improvement made by cross-matching. In fact, Figure 24 shows the distribution of the differences in accuracy between experiment one and two per number of faces, which tends to follow a right-skewed distribution. At both edges of the distribution, cross-matching can even degrade the initial accuracy. A reason for the unexpected contribution of the number of images per profile is that for profiles with only a few images, the algorithm fails to find sufficient candidates to be considered TCIs. Whereas in case of a large number of images, the number of TCIs in the DFS that are in reality the UFF (false negatives) gets large and directly impacts the cross-matching process by applying unwanted attraction of type three.

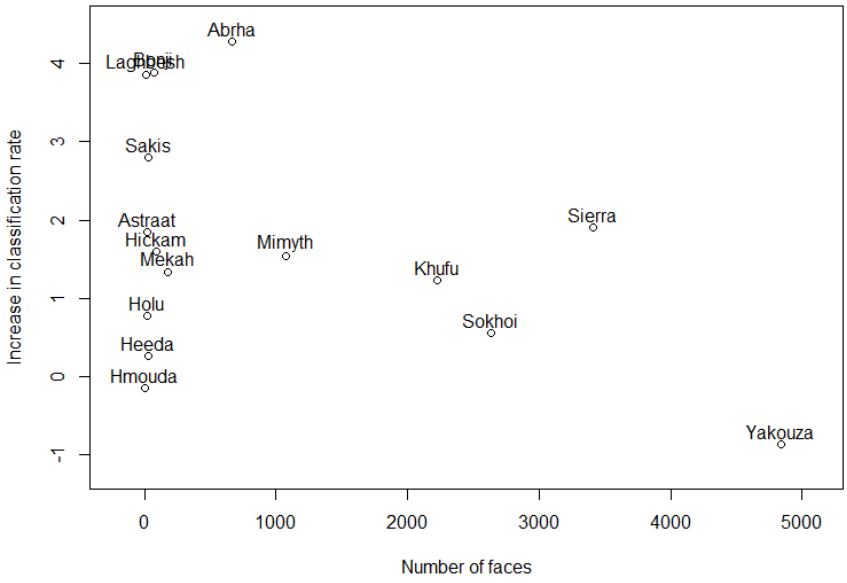


Figure 24. Distribution of Accuracy per Number of Observations.

C. EXPERIMENT 3: WEIGHTED CROSS-MATCHING

In this final stage, we apply the algorithm described in Chapter III and record the results for comparison. Table 12 is a review of experiment two alongside with the results of experiment three. The left arrows reflect the increase/decrease against the FVO experiment, whereas the right arrow is related to the progress made by experiment three against experiment two: weighted cross-matching versus cross-matching.

Table 12. Summary of Experiment 3.

Profiles	Experiment 2				Experiment 3			
Abrha	TP	FN	TN	FP	TP	FN	TN	FP
	472	195	24,186	2,262	486	181	25,607	841
Error rate	29.24%		8.55%		27.14%		3.18%	
F-Score	↑ **0.27757**				↑ **0.48746** ↑			
Accuracy	↑ **90.94%**				↑ **96.23%** ↑			
BIA	↑ **81.11%**				↑ **84.84%** ↑			
Astraat	TP	FN	TN	FP	TP	FN	TN	FP
	25	1	25,197	1,892	25	1	26,818	271
Error rate	3.85%		6.98%		3.85%		1.00%	
F-Score	↑ **0.02573**				↑ **0.15528** ↑			
Accuracy	↑ **93.02%**				↑ **99.00%** ↑			
BIA	↓ **94.58%**				↑ **97.58%** ↑			
Bonji	TP	FN	TN	FP	TP	FN	TN	FP
	59	11	26,588	457	57	13	26,920	125
Error rate	15.71%		1.69%		18.57%		0.46%	
F-Score	↑ **0.20137**				↑ **0.45238** ↑			
Accuracy	↑ **98.27%**				↑ **99.49%** ↑			
BIA	↑ **91.30%**				↑ **90.48%** ↓			
Heeda	TP	FN	TN	FP	TP	FN	TN	FP
	20	13	22,438	4,644	21	12	23,035	4,047
Error rate	39.39%		17.15%		36.36%		14.94%	
F-Score	↓ **0.00852**				↑ **0.01024** ↑			
Accuracy	↑ **82.83%**				↑ **85.03%** ↑			
BIA	↓ **71.73%**				↓ **74.35%** ↑			
Hickam	TP	FN	TN	FP	TP	FN	TN	FP
	74	14	26,512	515	63	25	26,929	98
Error rate	15.91%		1.91%		28.41%		0.36%	
F-Score	↑ **0.21861**				↑ **0.50602** ↑			
Accuracy	↑ **98.05%**				↑ **99.55%** ↑			
BIA	↓ **91.09%**				↓ **85.61%** ↓			
Hmouda	TP	FN	TN	FP	TP	FN	TN	FP
	5	0	24,559	2,551	5	0	24,646	2,464
Error rate	0.00%		9.41%		0.00%		9.09%	
F-Score	↑ **0.00390**				↑ **0.00404** ↑			
Accuracy	↑ **90.59%**				↑ **90.91%** ↑			
BIA	↑ **95.30%**				↑ **95.46%** ↑			

Table 12. Summary of Experiment 3 (Cont.).

Profiles	Experiment 2				Experiment 3			
Holu	TP	FN	TN	FP	TP	FN	TN	FP
	18	1	26,907	189	14	5	27,079	17
Error rate	5.26%		0.70%		26.32%		0.06%	
F-Score	↑ **0.15929**				↑ **0.56000** ↑			
Accuracy	↑ **99.30%**				↑ **99.92%** ↑			
BIA	↑ **97.02%**				↓ **86.81%** ↓			
Khufu	TP	FN	TN	FP	TP	FN	TN	FP
	2,000	227	24,363	525	2,026	201	24,331	557
Error rate	10.19%		2.11%		9.03		2.24	
F-Score	↑ **0.84175**				↑ **0.84241** ↑			
Accuracy	↑ **97.23%**				↑ **97.20%** ↓			
BIA	↑ **93.85%**				↑ **94.37%** ↑			
Laghbesh	TP	FN	TN	FP	TP	FN	TN	FP
	12	2	26,513	588	10	4	27,036	65
Error rate	14.29%		2.17%		28.57%		0.24%	
F-Score	↑ **0.03909**				↑ **0.22472** ↑			
Accuracy	↑ **97.82%**				↑ **99.75%** ↑			
BIA	↑ **91.77%**				↓ **85.59%** ↓			
Mekah	TP	FN	TN	FP	TP	FN	TN	FP
	153	26	26,686	250	150	29	26,840	96
Error rate	14.53%		0.93%		16.20%		0.36%	
F-Score	↑ **0.52577**				↑ **0.70588** ↑			
Accuracy	↑ **98.98%**				↑ **99.54%** ↑			
BIA	↑ **92.27%**				↓ **91.72%** ↓			
Mimyth	TP	FN	TN	FP	TP	FN	TN	FP
	960	114	25,269	772	889	185	25,633	408
Error rate	10.61%		2.96%		17.23%		1.57%	
F-Score	↑ **0.68425**				↑ **0.71989** ↑			
Accuracy	↑ **96.73%**				↑ **97.81%** ↑			
BIA	↑ **93.21%**				↓ **90.60%** ↓			
Sakis	TP	FN	TN	FP	TP	FN	TN	FP
	23	6	21,083	6,003	24	5	20,876	6,210
Error rate	20.69%		22.16%		17.24%		22.93%	
F-Score	↑ **0.00760**				↑ **0.00766** ↑			
Accuracy	↑ **77.84%**				↑ **77.08%** ↓			
BIA	↓ **78.57%**				↑ **79.92%** ↑			

Table 12. Summary of Experiment 3 (Cont.).

Profiles	Experiment 2				Experiment 3			
Sierra	TP	FN	TN	FP	TP	FN	TN	FP
	2,928	484	21,049	2,654	3,064	348	20,734	2,969
Error rate	14.19%		11.20%		10.20%		12.53%	
F-Score	↑ **0.65110**				↑ **0.64881** ↓			
Accuracy	↑ **88.43%**				↑ **87.77%** ↓			
BIA	↑ **87.31%**				↑ **88.64%** ↑			
Sokhoi	TP	FN	TN	FP	TP	FN	TN	FP
	2,322	312	16,548	7,933	2,443	191	15,017	9,464
Error rate	11.85%		32.40%		7.25%		38.66%	
F-Score	↑ **0.36031**				↓ **0.33602** ↓			
Accuracy	↑ **69.59%**				↓ **64.39%** ↓			
BIA	↓ **77.88%**				↓ **77.05%** ↓			
Yakouza	TP	FN	TN	FP	TP	FN	TN	FP
	4,543	293	17,370	4,909	4,662	174	16,042	6,237
Error rate	6.06%		22.03%		3.60%		27.99%	
F-Score	↓ **0.63592**				↓ **0.59256** ↓			
Accuracy	↓ **80.82%**				↓ **76.36%** ↓			
BIA	↑ **85.95%**				↓ **84.20%** ↓			
Global	TP	FN	TN	FP	TP	FN	TN	FP
	13,614	1,699	355,268	36,144	13,939	1,374	357,543	33,869
Error rate	11.10%		9.23%		8.97%		8.65%	
F-Score	↑ **0.41844**				↑ **0.44166** ↑			
Accuracy	↑ **90.70%**				↑ **91.33%** ↑			
BIA	↑ **89.84%**				↑ **91.19%** ↑			

We note a significant improvement in the global performance of face verification when combined with the metadata via weighted cross-matching. Table 13 is a summary of the evolution of the parameters across the three experiments reflecting the impact of cross-matching and weighted cross-matching.

Table 13. Global Summary of WCM against FV.

Parameters	Experiment 1	Experiment 2	Experiment 3
True Positives	13,381	13,614	13,939
True Negatives	346,878	355,268	357,543
False Positives	1,932	1,699	1,374
False Negatives	44,534	36,144	33,869
Class “1” Accuracy	87.38%	88.90%	91.03%
Class “0” Accuracy	88.62%	90.77%	91.35%
Balanced Inter-class Accuracy	88.00%	89.84%	91.19%
Accuracy	88.58%	90.70%	91.33%
F-Score	0.36546	0.41844	0.44166

The number of affected images by the WCM is relatively low compared to the dataset’s size. In fact, in order to demonstrate an improvement caused by WCM, we need to have two images where we detect the faces and succeed to properly identify one on the first image and then correlate the same person’s face on the second image using the metadata and a lenient face verification threshold. The number of pairs satisfying such conditions is not expected to be large, which explains the relatively minor improvement. Nevertheless, we succeed in taking advantage of preexisting data to refine results generated by the simple FV process and cross-matching. Figure 25 reflects the progress achieved by CM and WCM compared to face verification.

Although the performance of this last experiment is above 90%, the usefulness of the overall framework is still questionable. If the intent is to provide the user with a list of suggestions for annotation, in the first experiment we would have succeeded in 23.10% of the cases. The precision increases to 27.36% in the second experiment and then becomes 29.16% in the final experiment. The WCM leads to an amelioration in the

precision by the order of 26.23%. This means that the user will have one correct suggestion for every 3.4 attempts instead of 4.3.

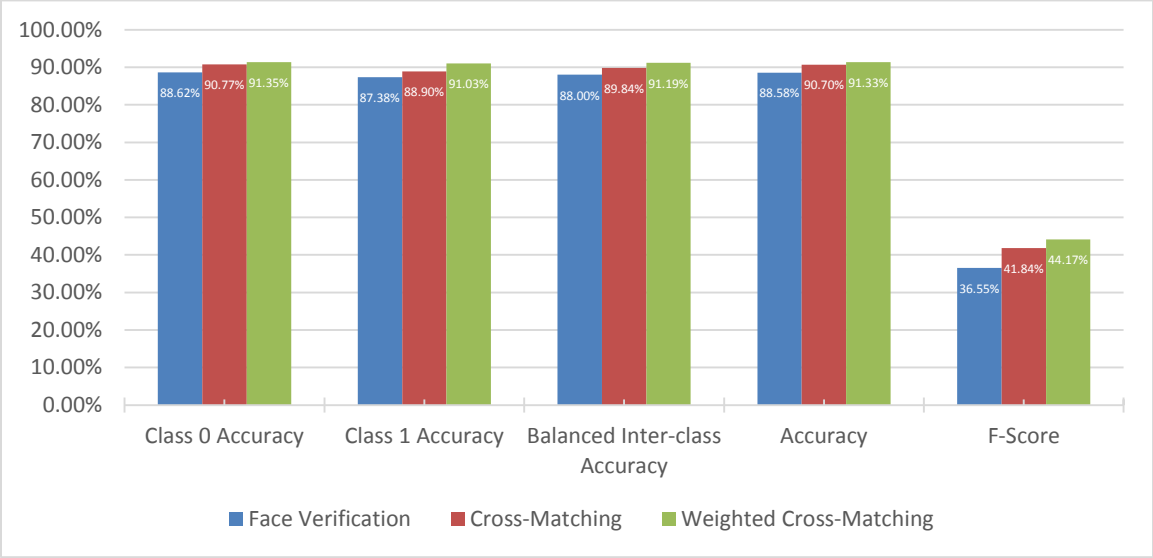


Figure 25. Comparison of the Three Experiments.

Running the third experiment on a PC (Pentium i7, 8GB RAM) takes four hours per profile. To better illustrate the reasons for such a slow process, we consider the example of the profile "Khufu." We have 27,115 faces in the dataset, so the algorithm has to go over the entire set to create the CFS, DFS, and CZ. We end up having this distribution:

- $|\text{CFS}| = 717$
- $|\text{DFS}| = 15{,}713$
- $|\text{CZ}| = 10{,}685$

We compare each element of the CZ against all the elements of the CFS and the DFS. This process makes $10{,}685 \times (15{,}713 + 717)$ iterations, thus $O(n^2)$. On a Hadoop cluster, the same operation takes roughly 3 minutes. The time measured does not include the face detection, alignment, and representation.

On a smart phone with 32GB of storage capacity, the average number of images is between 1,500 and 3,500 images. This means the number of faces will also be reduced compared to our dataset. When we execute the algorithm on the same laptop but on a sample of 2,000 faces, the execution time drops to less than 50 seconds.

V. CONCLUSION

Many approaches address the performance of face verification in an unconstrained environment, but most of them focus on improving the classification based on the content or the context. OpenFace from [12] gives a distance measure between two faces that can be used in many ways to improve the efficiency of face verification or for many other purposes, such as clustering. In this work, we use OpenFace to perform cross-matching that proved to be efficient on our dataset.

The intent of this research is to investigate whether additional refining of the results can be achieved without deploying heavy tools like deep neural networks so the implementation can fit on portable devices and provide the user better experience than face verification alone. The results revealed in the third experiment are promising, and our algorithm is not so sophisticated that it cannot fit on smartphones, cameras, and laptops. It is possible for cloud-service providers to adapt our framework to help improve their management of image files.

Although the precision from the three experiments performed in our work appears to be relatively low, we must take into consideration the nature of the dataset and the studied subjects. The profiles studied show a great variability in the cardinalities of the classes. Having a large number of true negatives causes the precision to drop. This issue is not likely to be observed in smaller datasets or mobile PDIL. While PDILs can grow large, incremental and continuous management helps compensate for the gap between the classes and yields meaningful correlation between subjects in regard of time and space.

The metadata plays a crucial role in data management, and it has been proven that it leads to improving face verification, even without creating per-user patterns. Most modern cameras are equipped with face detection capability. Including the location of the faces on the image and perhaps a representation in the metadata fields will reduce the load of face detection and annotation. Devices like Samsung smartphones (using Android version 7 or higher) or Apple iPhones (iOS 10 and above) record a short video at the time of capture that can be used to refine the face representation by opting for the least

unconstrained face among the frames of that flash video, whereas the image itself is just the last frame of the recording and provides only one ultimate choice to be used.

The standard benchmark LFW and other datasets from Flickr are used by researchers to address unconstrained faces and evaluate the performance of their frameworks against other people's works. However, there is no publically available PDIL (or representation) that can be used as reference by researchers. Privacy is a serious issue in this field, although the objective is to provide the users with a comfortable experience while managing their PDILs, leading to an optimized resource management. We provide the first publically available PDIL. All files and the entire dataset, including the python code, MapReduce scripts, and SQL commands can be found in our GitHub repository available at https://github.com/touwereg/Weighted-Cross-Matching/.

For future work, drawing on the OpenFace concept that measures the similarity of two faces, we address the possibility of creating a framework that first detects the conditions of the face to verify and simulate the same conditions on the reference face and then compares the faces again to determine the decision. For example, while checking a face, if this face appears to be covered by sunglasses, the algorithm should be able to apply the same mask to the original face, regenerate a representation, and then re-compare and evaluate the new output. Another approach is to combine all the existing techniques into one robust tool that addresses different patterns (expression, hair, and pose) and exploits all the cues in order to create a quasi-complete tool while considering the scalability of the algorithm and the tradeoff between execution time and accuracy. Finally, we suggest the possibility of using LIDAR (Light Detection and Ranging) in face verification. 3D extraction and face alignment are routine operations on LIDAR objects, and the face capture does not rely on lighting. The brightness and the image resolution are now shifted to the fidelity of the LIDAR point cloud, which relies on the number of sensors, the angle between the projected rays, and the distance from the source of light to the subject. The LIDAR is capable of capturing a 2D digital image along with the 3D object, which helps correlate a 2D to a 3D face and selects the best frame for further analysis based on the 3D prototype selection.

LIST OF REFERENCES

[1] "Slowing growth ahead for worldwide Internet audience," *eMarketer*, June 7, 2016. [Online]. Available: https://www.emarketer.com/Article/Slowing-Growth-Ahead-Worldwide-Internet-Audience/1014045

[2] R. Sarvas, E. Herrarte, A. Wilhelm, and M. Davis, "Metadata creation system for mobile images," in *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, 2004, pp. 36–48.

[3] W. Wagenaar, "My memory: A study of autobiographical memory over six years," in *Cognitive Psychology*, 1986, pp. 18:225–252.

[4] R. Saini and N. Rana, "Comparison of various biometric methods," *International Journal of Advances in Science and Technology*, vol. 2, no. 1, pp. 24–30, 2014.

[5] N. Zhang, M. Paluri, Y. Taigman, R. Fergus, and L. Bourdev, "Beyond frontal faces: Improving person recognition using multiple cues," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4804–4813.

[6] M. A. Turk and A. P. Pentland, "Face recognition using Eigenfaces," *Journal of Cognitive Neuroscience*, vol. 3, no. 1. pp. 71–86, 1991.

[7] N. Kumar, A. Berg, P. N. Belhumeur, and S. Nayar, "Describable visual attributes for face verification and image search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1962–1977, 2011.

[8] M. Welling, "Fisher linear discriminant analysis," *Department of Computer Science Technical Report, University of Toronto*, vol. 3, no. 1, 2005.

[9] S. U. Hussain, T. Napoléon, and F. Jurie, "Face recognition using local quantized patterns," in *British Machine Vision Conference*, 2012, p. 11.

[10] H. J. Seo and P. Milanfar, "Face verification using the Lark representation," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1275–1286, 2011.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[12] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," *CMU-CS-16-118, CMU Sch. Comput. Sci. Tech. Rep.*, 2016.

[13] A. Rutkin, “Facebook can recognise you in photos even if you’re not looking,” *New Scientist Daily*, Jun. 22, 2015. [Online]. Available: https://www.newscientist.com/article/dn27761-facebook-can-recognise-you-in-photos-even-if-youre-not-looking/

[14] C. D. Castillo and D. W. Jacobs, “Using stereo matching for 2-d face recognition across pose,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[15] V. Blanz, S. Romdhani, and T. Vetter, “Face identification across different poses and illuminations with a 3d morphable model,” in *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 202–207.

[16] G. Hua and A. Akbarzadeh, “A robust elastic and partial matching metric for face recognition,” in *12th IEEE International Conference on Computer Vision*, 2009, pp. 2082–2089.

[17] J. Roth and X. Liu, “On hair recognition in the wild by machine,” in *28th AAAI Conference on Artificial Intelligence*, 2014, pp. 2824–2830.

[18] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, “Attribute and simile classifiers for face verification,” in *12th IEEE International Conference on Computer Vision*, 2009, pp. 365–372.

[19] H. Li, J. Brandt, Z. Lin, X. Shen, and G. Hua, “A multi-level contextual model for person recognition in photo albums,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1297–1305.

[20] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[21] R. B. Yeh, A. Paepcke, H. Garcia-Molina, and M. Naaman, “Leveraging context to resolve identity in photo albums,” in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2005, pp. 178–187.

[22] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” *Univ. Massachusetts Amherst Tech. Rep.*, vol. 1, pp. 07–49, 2007.

[23] L. Bourdev and J. Malik, “Poselets: Body part detectors trained using 3d human pose annotations,” in *12th IEEE International Conference on Computer Vision*, 2009, pp. 1365–1372.

[24] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *12th IEEE International Conference on Computer Vision*, 2009, pp. 498–505.

[25] L. Zhang, L. Chen, M. Li, and H. Zhang, "Automated annotation of human faces in family albums," in *Proceedings of the 11th ACM International Conference on Multimedia*, 2003, pp. 355–358.

[26] M. Ingram, "Facebook's new algorithm can recognize you even if your face is hidden," *Fortune*, Jun. 23, 2015. [Online]. Available: http://fortune.com/2015/06/23/facebook-facial-recognition/

[27] P. I. Wilson and J. Fernandez, "Facial feature detection using Haar classifiers," *Journal of Computing Sciences in Colleges*, vol. 21, no. 4, pp. 127–133, 2006.

[28] D. Anguelov, K. Lee, S. B. Gokturk, and B. Sumengen, "Contextual identity recognition in personal photo albums," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–7.

[29] M. Davis, S. King, N. Good, and R. Sarvas, "From context to content: leveraging context to infer media metadata," in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, 2004, pp. 188–195.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol.12, pp. 2825-2830, 2011

[31] J. Beall, "How Google uses metadata to improve search results," *The Serials Librarian*, vol. 59, no. 1, pp. 40–53, 2010.

[32] T. Simonite, "Facebook creates software that matches faces almost as well as you do," *MIT Technology Review*, March 17, 2014. [Online]. Available: https://www.technologyreview.com/s/525586/facebook-creates-software-that-matches-faces-almost-as-well-as-you-do/

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California